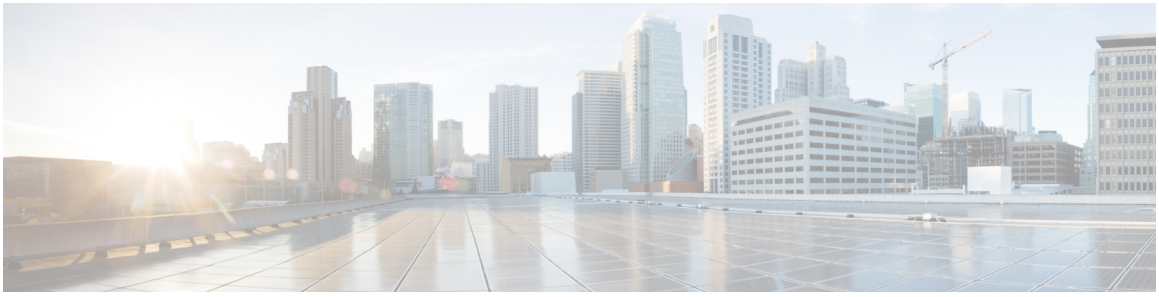
IP SLAs Configuration Guide

First Published: 2025-09-15

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

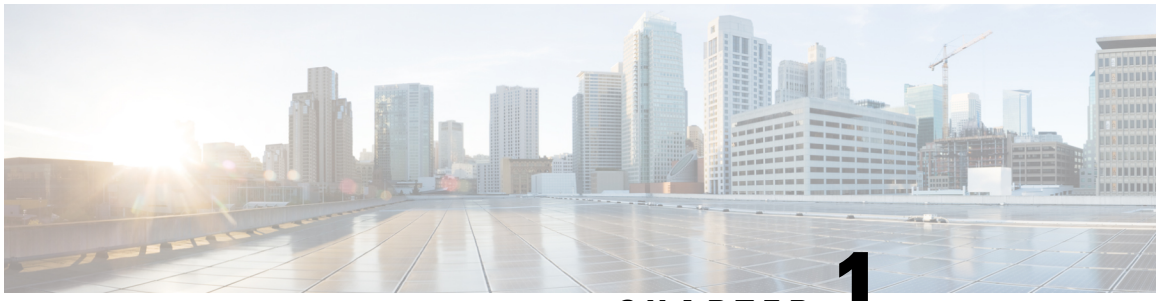
Read Me First



Read Me First

Only supported features are documented. To confirm or clarify all the supported features for a platform, go to Cisco Feature Navigator.

CONTENTS



CHAPTER 1

IP SLAs

Feature History for IP SLA

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	IP SLAs: Cisco IP SLAs is a network enhancement feature that enables proactive network performance measurement by sending test data across the network to monitor performance between multiple locations or paths.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

Cisco IP SLAs

Cisco IP Service Level Agreements (IP SLAs) is a network enhancement feature that

- enables proactive network performance measurement

- by sending test data across the network
- to monitor performance between multiple locations or paths.

How does the IP SLA work

Summary

IP SLA operates by creating and sending test traffic across your network, mimicking real user data or application flows. This process helps you measure how well your network is performing in real time.

Workflow

1. IP SLA generates test packets that behave like actual network or application traffic (for example, voice, video, or web traffic).
2. These packets are sent between two network devices (such as routers or switches) or from a Cisco device to another IP-enabled device, like a server or another network appliance.
3. The receiving device (or application server) processes the test packets and may respond, depending on the type of IP SLA operation. The sending device measures how long the process takes, how many packets arrive, and other key performance metrics.
4. By analyzing this test traffic, IP SLA provides real-time data about network health, such as latency, jitter, packet loss, and connectivity.
5. These measurements help network administrators quickly detect, troubleshoot, and resolve network issues. The data is also valuable for analyzing network performance trends and for designing or optimizing network topologies.

Are IP SLAs restricted by network topology?

Cisco IP SLAs operate at the IP layer (Layer 3 of the OSI model), which means they do not rely on the underlying Layer 2 (data link layer) transport technologies such as Ethernet, Frame Relay, or MPLS. As a result, IP SLAs can be set up to send test traffic across any type of network infrastructure.

The benefits of this are:

- End-to-End monitoring

  You can measure network performance from one end of the network to another, regardless of how the devices are connected or what types of underlying connections are used.

- Works across diverse networks

  IP SLAs work over any combination of network types—wired, wireless, WAN, LAN, VPN, etc.

- User experience focus

  Because IP SLAs test traffic takes the same path as actual user traffic, the metrics collected (like delay, packet loss, or jitter) accurately reflect what end users are experiencing.

Performance metrics for IP SLAs

Cisco IP SLAs gather a variety of important network performance metrics. These metrics help network administrators understand how well the network is operating and how users experience network services. Here's what each metric means:

- Round-trip and one-way Delay

  Measures how long it takes for data to travel from the source to the destination and back (round-trip), or just one way. This helps determine if there are delays in the network that could affect applications.

- Directional jitter

  Measures the variation in delay between packets as they travel in one direction. High jitter can cause problems for real-time applications like voice and video.

- Directional packet loss

  Tracks how many packets are lost between the source and destination in one direction. Packet loss can result in poor application performance or dropped calls.

- Packet sequencing (order of arrival)

  Checks whether packets arrive in the same order they were sent. Out-of-order packets can disrupt certain applications, especially voice and video.

- Per-hop path information

  Provides details about each step (hop) a packet takes between source and destination. This can help identify where in the network problems are occurring.

- Directional connectivity

  Verifies if a path is up and reachable in a specific direction, helping to detect outages or unreachable segments.

- Server or website download times

  Measures how long it takes to download content from a server or website, simulating the user's experience when accessing online resources.

Collecting and analyzing these metrics allows network administrators to monitor, troubleshoot, and optimize network performance. By understanding the quality of the network from the perspective of the end user, they can ensure reliable and satisfactory service for critical applications and services.

SNMP and IP SLA

Cisco IP SLAs can send their performance data using a standard protocol called SNMP (Simple Network Management Protocol). Because SNMP is widely supported, this allows IP SLA measurements to be collected and displayed by many network monitoring and management applications, such as Cisco Prime IPM and other third-party tools.

SNMP with IP SLA provides these benefits:

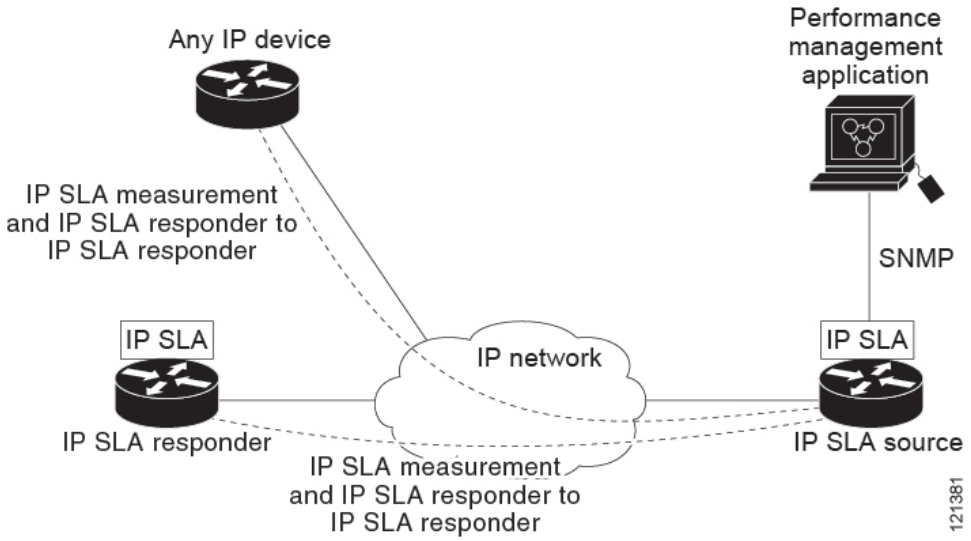
- Network administrators can see all the IP SLA data in one place, rather than checking each device individually. Efficient Troubleshooting: With all performance data available centrally, it's faster and easier to identify and resolve network problems.
- Management tools can analyze trends, create reports, and display performance metrics graphically, making it easier to spot issues and understand overall network health.
- With all performance data available centrally, it's faster and easier to identify and resolve network problems.

Network performance measurement with Cisco IP SLAs

Cisco IP SLAs help you monitor how well your network is performing across any part of your network—from the core (central routers and switches), to the distribution layer, and out to the network edge (branch offices or remote sites). Unlike traditional monitoring that might require extra hardware devices ("probes") placed throughout the network, IP SLAs are built into Cisco devices, so no additional equipment is needed.

The following figure shows how IP SLAs begin when the source device sends a generated packet to the destination device. After the destination device receives the packet, depending on the type of IP SLAs operation, it responds with time-stamp information for the source to make the calculation on performance metrics. An IP SLAs operation performs a network measurement from the source device to a destination in the network using a specific protocol such as UDP.

Figure 1: IP SLAs Operations



IP SLA responder and IP SLA control protocol

The IP SLA responder is a special software feature built into Cisco devices (such as routers or switches) that acts as the target for IP SLA test packets. When you run an IP SLA test, the responder is enabled on the destination device so it can recognize these specific test packets and respond to them.

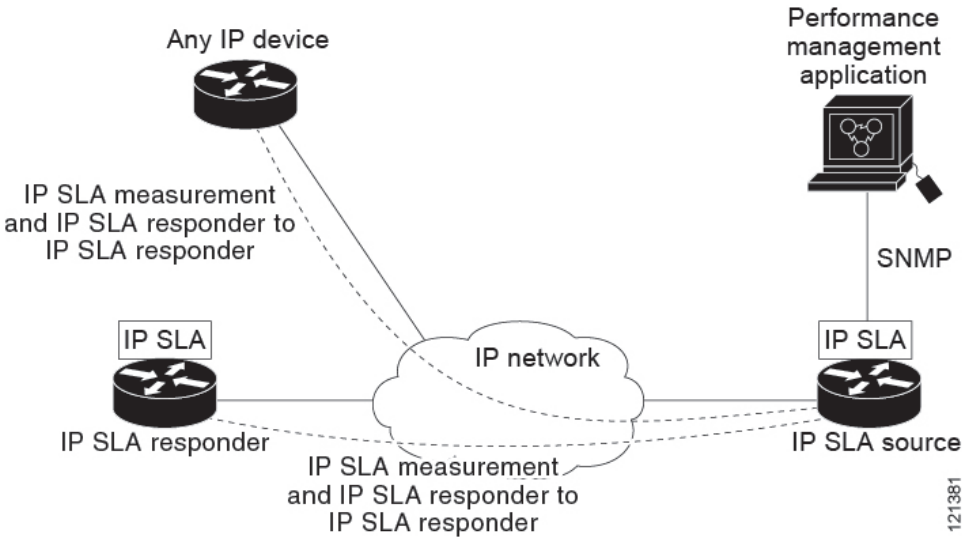
The responder helps provide very accurate measurements of network performance, such as delay or packet loss, because it processes the test packets quickly and can add precise time-stamp information. Since the

responder is built into Cisco devices, you don't need to deploy or buy any extra hardware (like dedicated probes).

The responder uses the Cisco IOS IP SLA Control Protocol, which tells the device exactly which port (a network communication endpoint) to listen on and for how long. This ensures the responder is only active when needed and on the correct port, improving both security and accuracy. The responder can be enabled on Cisco devices operating at Layer 2 (the data link layer), and it doesn't need to support all IP SLA features—just the ability to recognize and reply to test packets.

The following figure shows where the Cisco IOS IP SLA responder fits in the IP network. The Cisco IOS IP SLA responder operates by listening on a designated port for control protocol messages that are sent by an IP SLA operation. When the responder receives a control message, it temporarily enables the specified UDP or TCP port for a set period of time. During this active window, the responder accepts incoming requests and replies to them, facilitating precise network measurements. Once it has responded to the IP SLA packet or the configured duration ends, the responder disables the port to maintain security and efficiency. For enhanced security, MD5 authentication can be used for the control messages, ensuring only authorized operations are processed.

Figure 2: Cisco IOS IP SLAs Operation



It is not always necessary to enable the responder on the destination device for every IP SLA operation. If the IP SLA test is targeting services that are already running on the destination device, such as Telnet or HTTP, the responder feature is not needed. In these scenarios, the IP SLA operation can interact directly with the existing service, which simplifies the configuration process and eliminates the need for extra setup steps on the destination device. This makes it easier and faster to deploy IP SLA monitoring for commonly used network services.

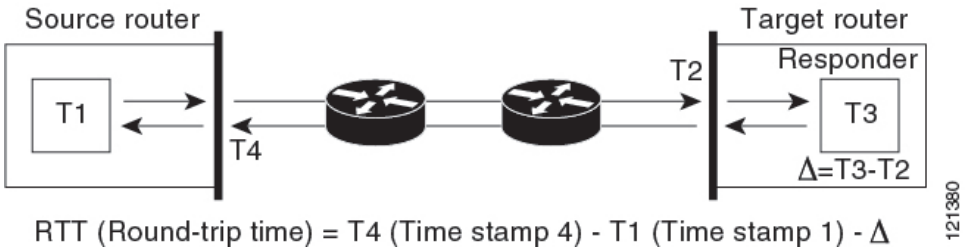
Response time computation for IP SLAs

When measuring network performance, it's important to get accurate response times. However, network devices like switches, controllers, and routers sometimes have to handle many tasks at once. This can cause short processing delays—so when a test packet arrives, it might wait in a queue before being processed or replied to. If these processing delays are included in the response time measurement, the results may not reflect the actual time it takes for data to travel across the network.

To overcome this, IP SLAs use precise time stamps to measure the exact moment a test packet enters and leaves a device. When the IP SLA responder feature is enabled, the device marks the time as soon as the packet arrives (at the interrupt level—before it's queued for processing) and again when it leaves. By subtracting out the time the packet spends being processed inside the device, IP SLAs provide a much more accurate measurement of the real network delay, not just how fast the device processes packets. This time stamping is done with very fine (sub-millisecond) accuracy, ensuring that even small delays are measured precisely.

The following figure demonstrates how the responder works. When the IP SLA responder is enabled, the round-trip time calculation becomes highly accurate by using four time stamps. As the test packet travels, the target router records the exact time it receives the packet (TS2) and the time it sends the response back (TS3). The difference between these two times, called delta, represents the processing time on the target device. This processing time is then subtracted from the total round-trip time to eliminate any delays caused by the device itself. Similarly, on the source router, the final arrival time of the response (TS4) is also captured at the interrupt level for maximum accuracy. By using this method, IP SLAs ensure that the measured round-trip time closely reflects only the actual network delay, not any internal device processing time.

Figure 3: Cisco IOS IP SLA Responder Time Stamping



Another important benefit of taking two time stamps at the target device is that it enables monitoring of advanced performance metrics like one-way delay, jitter, and directional packet loss. This is valuable because network traffic can often behave differently in each direction, so having detailed, directional statistics provides a more accurate picture of network health. To measure one-way delay accurately, both the source and target routers must have their clocks synchronized, typically using the Network Time Protocol (NTP). However, one-way jitter can still be measured even if the clocks are not synchronized, which allows administrators to assess variations in packet transit times without needing exact time alignment between devices. This flexibility makes it easier to analyze and troubleshoot network performance under real-world conditions.

Monitoring and storing network performance statistics

When you set up IP SLA operations on a Cisco device, the device continuously monitors various network performance statistics—such as delay, jitter, and packet loss—based on the type of test you configure. The statistics are saved directly on the Cisco device that is performing the IP SLA operation.

There are two options available for monitoring and storing network performance statistics. They are:

- You can use Cisco IOS commands to view the results directly from the device's console or terminal.
- The data is also available via SNMP, a standard network management protocol. This means you can use network monitoring tools to automatically collect, analyze, and display the performance statistics from multiple devices in a centralized system.

Customization of IP SLA packets

IP SLA packets can be customized with different IP and application layer options means that when you set up an IP SLA operation (a network test), you can adjust several parameters of the test packets. This customization helps you make the test simulate real network conditions or target specific parts of your network.

The options you can customize include:

- Source and Destination IP Addresses

  You can choose which device (IP address) sends the test packets and which device receives them.

- UDP or TCP Port Numbers

  You can specify which application port to use (for example, port 80 for HTTP or port 5060 for VoIP/SIP) so you can test the path for specific applications or services.

- Type of Service (ToS) Byte Settings

  This includes settings like DSCP and IP Precedence, which are used for Quality of Service (QoS). This lets you see how high-priority or low-priority traffic performs across your network.

- VRF Instances

  If your network uses VPNs or multiple routing tables, you can specify which VPN or VRF context to use for the test, ensuring you're measuring performance for the correct network segment.

- URL Web Addresses

  You can target a specific website or web application to test if it's reachable and how it performs from your network's perspective.

Network administrators often need to monitor different traffic types and paths. With these customization options, they can:

- Test exactly the traffic their users or applications use.
- Simulate real-life network scenarios for accurate measurement.
- Verify the performance of specific business-critical services or applications.
- Ensure the network meets required service levels for different traffic types or customers.

Benefits of IP SLA

Using IP SLAs offers several important benefits for network management and monitoring:

- Service-Level Agreement (SLA) monitoring and verification

  IP SLAs help you track and verify whether your network is meeting the performance standards promised in SLAs with service providers or internal customers. This means you can prove the network is delivering the agreed level of service.

- Comprehensive network performance monitoring

  By measuring critical metrics like jitter, latency, and packet loss, IP SLAs provide a detailed and ongoing view of network health. These reliable and predictable measurements make it easier to assess network performance over time.

- Quality of Service (QoS) assessment

  IP SLAs allow you to check if your current network configuration can support new IP-based services (like voice or video), and whether your QoS settings are effective. This is important before rolling out new applications that require certain performance levels.

- End-to-End Network Availability

  You can use IP SLAs to test network connections from one end to the other (edge to edge), including remote sites. This is useful for confirming that important resources, like servers storing business data, are reachable at all times.

- Efficient Troubleshooting

  IP SLAs provide consistent and accurate performance data, helping you quickly pinpoint and fix network issues. This saves time and reduces the impact of problems on users and business operations.

- Support for MPLS Networks

  On devices that use MPLS, IP SLAs can also measure and verify the performance of MPLS paths, ensuring that these advanced network segments meet business requirements.

Configure the IP SLA responder

The IP SLA responder is available only on Cisco IOS software-based devices, including some Layer 2 devices that do not support full IP SLA functionality.

Perform this task to configure the IP SLA responder on the target device (the operational target)

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla responder** {**tcp-connect** | **udp-echo**} **ipaddress** *ip-address* **port** *port-number*

Example:

```
Device(config)# ip sla responder udp-echo 172.29.139.134 5000
```

Configures the device as an IP SLA responder.

- **tcp-connect**: Enables the responder for TCP connect operations.

- **udp-echo**: Enables the responder for User DatagramProtocol (UDP) echo or jitter operations.
- **ipaddress** *ip-address*: Enter the destination IP address.
- **port** *port-number*: Enter the destination port number.

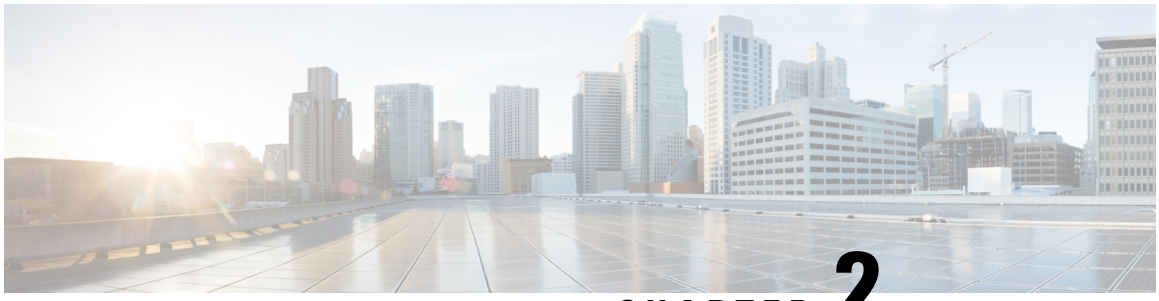
Note
The IP address and port number must match those configured on the source device for the IP SLA operation.

Step 4 **end**

Example:

```
Device(config)# end
```

Exits to privileged EXEC mode.



CHAPTER 2

IP SLAs DHCP Operation

Feature History for IP SLAs - DHCP Operation

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	IP SLAs - DHCP Operation: This operation measures the round-trip time (RTT) needed to discover a DHCP server and obtain a leased IP address.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

IP SLAs DHCP

The IP SLAs DHCP operation measures the round-trip time (RTT) needed to discover a DHCP server and obtain a leased IP address. After the operation completes, IP SLAs releases the leased IP address. The RTT data collected from this process can be used to assess DHCP server performance and ensure efficient IP address allocation within the network.

IP SLAs DHCP relay agent options

A DHCP relay agent is a host that forwards DHCP packets between clients and servers, enabling communication when clients and servers are not on the same physical subnet. Relay agents are essential in networks where direct communication is not possible due to subnet boundaries. Unlike standard IP packet forwarding, which switches packets transparently between networks, a relay agent receives DHCP messages and generates new

DHCP messages to send out on another interface, ensuring that DHCP requests and replies reach their intended destinations.

How IP SLAs DHCP works

The DHCP operation supports two modes. By default, it sends discovery packets on every available IP interface on the device to locate a DHCP server. However, if a specific DHCP server is configured, the operation directs discovery packets only to that designated server. This flexibility allows for either broad network testing or targeted performance monitoring of a particular DHCP server.

Configure a DHCP operation on the source device

Follow the steps in each of these tasks to configure a DHCP operation on the source device.

Before you begin

There is no need to configure an IP SLAs responder on the destination device.

Procedure

Step 1 Configure any one of these tasks

- Configure a basic DHCP operation
- Configure a DHCP operation with optional parameters

Step 2 Schedule IP SLAs operations

Configure a basic DHCP operation

Perform this task to configure a basic DHCP operation.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla** *operation-number*

Example:

```
Device(config)# ip sla 10
```

Starts configuring an IP SLAs operation and enters IP SLA configuration mode.

Step 4 **dhcp** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*}]

Example:

```
Device(config-ip-sla)# dhcp 10.10.10.3
```

Defines a DHCP operation and enters IP SLA DHCP configuration mode.

- *destination-ip-address*: The IP address of the DHCP server to which the discovery packet will be sent.
- *destination-hostname*: The hostname of the DHCP server to which the discovery packet will be sent.
- **source-ip** {*ip-address* | *hostname*}: (Optional) Specifies the source IP address or hostname from which the DHCP request will be sent.

Step 5 **frequency** *seconds*

Example:

```
Device(onfig-ip-sla-dhcp)# frequency 90
```

(Optional) Sets the rate at which a specified IP SLAs operation repeats.

Step 6 **end**

Example:

```
Device(config-ip-sla-dhcp)# end
```

Exits to privileged EXEC mode.

Configure a DHCP operation with optional parameters

Perform this task to configure a DHCP operation with optional parameters.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla** *operation-number*

Example:

```
Device(config)# ip sla 10
```

Starts configuring an IP SLAs operation and enters IP SLA configuration mode.

Step 4 **dhcp** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*}]

Example:

```
Device(config-ip-sla)# dhcp 10.10.10.3
```

Defines a DHCP operation and enters IP SLA DHCP configuration mode.

- *destination-ip-address*: The IP address to query for its host name (reverse lookup).
- *destination-hostname*: The host name to query for its IP address (standard lookup).
- **name-server** *ip-address*: The IP address of the DHCP server to be used for the query.
- **source-ip** {*ip-address* | *hostname*}: (Optional) Specifies the source IP address or hostname for the request.
- **source-port** *port-number*: (Optional) Specifies the source port number for the DHCP request.

Step 5 **history buckets-kept** *size*

Example:

```
Device(config-ip-sla-dhcp)# history buckets-kept 25
```

(Optional) Sets the number of history buckets that are kept during the lifetime of an IP SLAs operation.

Step 6 **history distributions-of-statistics-kept** *size*

Example:

```
Device(config-ip-sla-dhcp)# history distributions-of-statistics-kept 5
```

(Optional) Sets the number of statistics distributions kept per hop during an IP SLAs operation.

Step 7 **history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*]

Example:

```
Device(config-ip-sla-dhcp)# history enhanced interval 900 buckets 100
```

(Optional) Enables enhanced history gathering for an IP SLAs operation.

- **interval** *seconds*: (Optional) The time interval, in seconds, between each statistics recording.
- **buckets** *number-of-buckets*: (Optional) The number of history data buckets (records) to store for the operation.

Step 8 **history filter** {**none** | **all** | **overThreshold** | **failures**}

Example:

```
Device(config-ip-sla-dhcp)# history filter failures
```

(Optional) Defines the type of information kept in the history table for an IP SLAs operation.

Step 9 **frequency** *seconds*

Example:

```
Device(config-ip-sla-dhcp)# frequency 30
```

(Optional) Sets the rate at which a specified IP SLAs operation repeats.

Step 10 **history hours-of-statistics-kept** *hours*

Example:

```
Device(config-ip-sla-dhcp)# history hours-of-statistics-kept 4
```

(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.

Step 11 **history lives-kept** *lives*

Example:

```
Device(config-ip-sla-dhcp)# history lives-kept 5
```

(Optional) Sets the number of lives maintained in the history table for an IP SLAs operation.

Step 12 **owner** *owner-id*

Example:

```
Device(config-ip-sla-dhcp)# owner admin
```

(Optional) Configures the Simple Network Management Protocol (SNMP) owner of an IP SLAs operation.

Step 13 **history statistics-distribution-interval** *milliseconds*

Example:

```
Device(config-ip-sla-dhcp)# history statistics-distribution-interval 10
```

(Optional) Sets the time interval for each statistics distribution kept for an IP SLAs operation.

Step 14 **tag** *text*

Example:

```
Device(config-ip-sla-dhcp)# tag TelnetPollServer1
```

(Optional) Creates a user-specified identifier for an IP SLAs operation.

Step 15 **threshold** *milliseconds*

Example:

```
Device(config-ip-sla-dhcp)# threshold 10000
```

(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.

Step 16 **timeout** *milliseconds*

Example:

```
Device(config-ip-sla-dhcp)# timeout 10000
```

(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.

Step 17 **end**

Example:

```
Device(config-ip-sla-dhcp)# end
```

Exits to privileged EXEC mode.

Schedule IP SLAs operations

Perform this task to schedule IP SLAs operations.

Before you begin

- All IP SLAs operations to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group must be the same.
- The list of one or more operation ID numbers to be added to a multioperation group must be limited to a maximum of 125 characters in length, including commas (,).

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh:mm:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*}] [**ageout** *seconds*] [**recurring**]

Example:

```
Device(config)# ip sla schedule 10 life forever start-time
OR
Device(config)# ip sla schedule 1 3,4,6-9 schedule-period 50 frequency range 80-100
```

(Optional) Configures the scheduling parameters for an individual IP SLAs operation.

- *operation-number*: The IP SLA operation number to schedule (must match a previously created IP SLA operation).

  The range is from 1 to 2147483647.

- **life** {**forever** | *seconds*}: How long the operation will run.
  - **forever**: Runs the operation continuously until manually stopped.

- *seconds*: Number of seconds the operation should run.

  The range is from 1 to 2147483647 seconds.

- **start-time** {[*hh:mm:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*}: Specifies when to start the operation.
  - *hh:mm:ss* [*month day* | *day month*]: Specific time and date.
  - **pending**: Waits for a manual start.
  - **now**: Starts immediately.
  - **after** *hh:mm:ss*: Starts after the specified amount of time.
- **ageout** *seconds*: Time (in seconds) after which the operation is automatically deleted.

  The range is from 0 to 2147483647 seconds.
- **recurring**: Makes the operation run repeatedly according to its frequency settin

Step 4 **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* |**schedule-together**}[**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm* [*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm*[*:ss*]}]

Example:

```
Device(config)# ip sla group schedule 10 schedule-period frequency
OR
Device(config)# ip sla group schedule 1 3,4,6-9 life forever start-time now
```

(Optional) Specifies an IP SLAs operation group number and the range of operation numbers for a multioperation scheduler.

- *group-operation-number*: The number assigned to the group operation (must be unique).

  The range is from 1 to 2147483647.
- *operation-id-numbers*: List of individual IP SLA operation numbers to be included in the group.

  The range is from 1 to 2147483647 (can be a series separated by spaces).
- **schedule-period** *schedule-period-range*: Schedules each operation in the group with a specified time period between them.

  The range is from 1 to 604800 (seconds; up to 7 days).
- **schedule-together**: Starts all operations in the group at the same time.
- **frequency** *group-operation-frequency*: How often (in seconds) the group operation runs.

  The range is from 1 to 604800 seconds.

Step 5 **end**

Example:

```
Device(config)# end
```

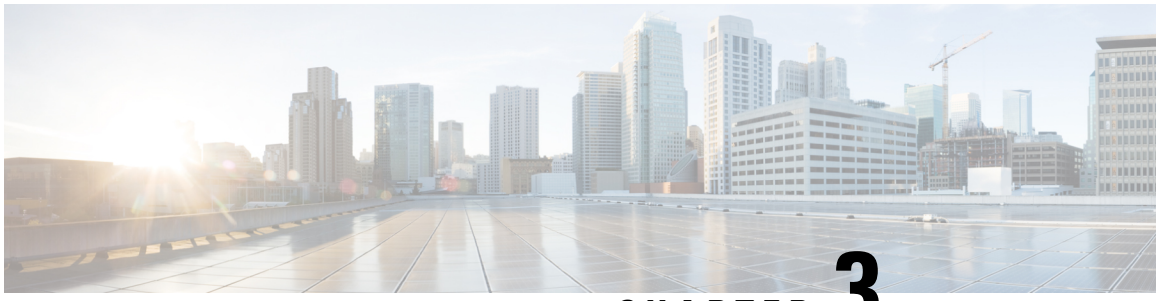
Exits global configuration mode and returns to privileged EXEC mode.

Configuration example for an IP SLAs DHCP operation

In the following example, IP SLAs operation number 12 is configured as a DHCP operation enabled for DHCP server 172.16.20.3. Note that DHCP option 82 is used to specify the circuit ID.

Device B Configuration

```
Device> enable
Device# configure terminal
Device(config)# ip dhcp-server 172.16.20.3
!
Device(config)# ip sla 12
Device(config-ip-sla)# dhcp 10.10.10.3
Device(config-ip-sla)# frequency 30
Device(config-ip-sla)# timeout 5000
Device(config-ip-sla)# tag DHCP_Test
Device(config-ip-sla)# exit
!
Device(config)# ip sla schedule 12 start-time now
```



CHAPTER 3

IP SLAs DNS Operation

Feature History for IP SLAs - DNS Operation

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	IP SLAs - DNS Operation: This operation measures the time difference between sending a DNS request and receiving a reply.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

IP SLAs DNS

The IP SLAs DNS operation is designed to measure the time difference between sending a DNS request and receiving a reply. By analyzing the results of the DNS operation, network administrators can determine the DNS lookup time, which is a crucial factor in assessing the performance of DNS or web servers. Monitoring DNS response times helps identify potential issues and ensures optimal performance for applications that rely on fast and reliable domain name resolution.

How IP SLAs DNS works

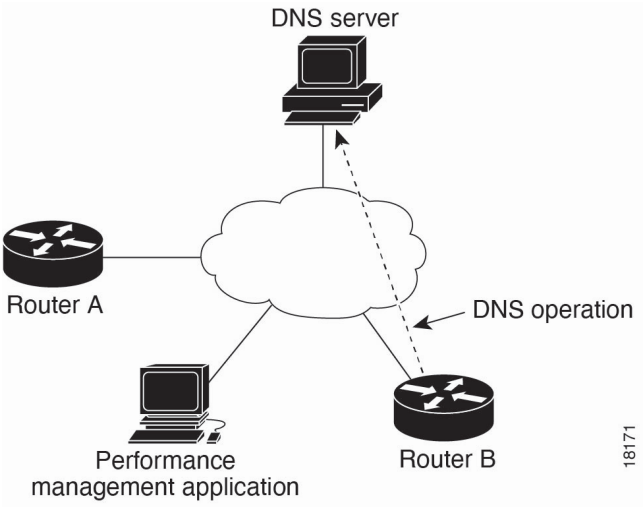
Summary

The DNS operation measures the time difference between sending a DNS request and receiving a reply, providing valuable insight into DNS server responsiveness. DNS plays a critical role on the Internet by translating network node names into IP addresses. With the IP SLAs DNS operation, a device can query for an IP address when a host name is provided, or query for a host name when an IP address is specified, allowing comprehensive monitoring of DNS resolution performance.

In the figure below Device B is configured as the source IP SLAs device and a DNS operation is configured with the DNS server as the destination device.

Workflow

Figure 4: DNS Operation



Connection response time is computed by measuring the difference between the time taken to send a request to the DNS server and the time a reply is received by Device B. The resulting DNS lookup time can help you analyze your DNS performance.

Faster DNS lookup times translate to a faster web server access experience.

Guidelines to to configure IP SLAs DNS

- If the IP Service Level Agreements (SLAs) operation is not running and not generating statistics, add the **verify-data** command to the configuration (while configuring in IP SLA configuration mode) to enable data verification. When data verification is enabled, each operation response is checked for corruption. Use the **verify-data** command with caution during normal operations because it generates unnecessary overhead.
- Use the **debug ip sla trace** and **debug ip sla error** commands to help troubleshoot issues with an IP SLAs operation.
- There is no need to configure an IP SLAs responder on the destination device.

Configure an IP SLAs DNS operation on the source device

Follow the steps in each of these tasks to configure an IP SLAs DNS operation on the source device.

Before you begin

There is no need to configure an IP SLAs responder on the destination device.

Procedure

Step 1 Perform any one of these tasks:

- Configure a basic DNS operation on the source device
- Configure a DNS operation with optional parameters on the source device

Step 2 Schedule IP SLAs operations

Configure a basic DNS operation on the source device

Perform this task to configure a basic DNS operation on the source device.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla** *operation-number*

Example:

```
Device(config)# ip sla 10
```

Starts configuring an IP SLAs operation and enters IP SLA configuration mode.

Step 4 **dns** {*destination-ip-address* | *destination-hostname*} **name-server** *ip-address* [**source-ip** {*ip-address* | *hostname*} **source-port** *port-number*]

Example:

```
Device(config-ip-sla)# dns host1 name-server 172.20.2.132
```

Defines a DNS operation and enters IP SLA DNS configuration mode.

Step 5 **frequency** *seconds*

Example:

```
Device(config-ip-sla-dns)# frequency 90
```

(Optional) Sets the rate at which a specified IP SLAs operation repeats.

Step 6 **end**

Example:

```
Device(config-ip-sla-dns)# end
```

Exits to privileged EXEC mode.

Configure a DNS operation with optional parameters on the source device

Perform this task to configure a DNS operation with optional parameters on the source device.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla** *operation-number*

Example:

```
Device(config)# ip sla 10
```

Starts configuring an IP SLAs operation and enters IP SLA configuration mode.

Step 4 **dns** {*destination-ip-address* | *destination-hostname*} **name-server** *ip-address* [**source-ip** {*ip-address* | *hostname*} **source-port** *port-number*]

Example:

```
Device(config-ip-sla)# dns host1 name-server 172.20.2.132
```

Defines a DNS operation and enters IP SLA DNS configuration mode.

- *destination-ip-address*: The IP address to query for its host name (reverse lookup).
- *destination-hostname*: The host name to query for its IP address (standard lookup).
- **name-server** *ip-address*: The IP address of the DNS server to be used for the query.
- **source-ip** {*ip-address* | *hostname*}: (Optional) Specifies the source IP address or hostname for the request.
- **source-port** *port-number*: (Optional) Specifies the source port number for the DNS request.

Step 5 **history buckets-kept** *size*

Example:

```
Device(config-ip-sla-dns)# history buckets-kept 25
```

(Optional) Sets the number of history buckets that are kept during the lifetime of an IP SLAs operation.

Step 6 **history distributions-of-statistics-kept** *size*

Example:

```
Device(config-ip-sla-dns)# history distributions-of-statistics-kept 5
```

(Optional) Sets the number of statistics distributions kept per hop during an IP SLAs operation.

Step 7 **history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*]

Example:

```
Device(config-ip-sla-dns)# history enhanced interval 900 buckets 100
```

(Optional) Enables enhanced history gathering for an IP SLAs operation.

- **interval** *seconds*: (Optional) The time interval, in seconds, between each statistics recording.
- **buckets** *number-of-buckets*: (Optional) The number of history data buckets (records) to store for the operation.

Step 8 **history filter** {**none** | **all** | **overThreshold** | **failures**}

Example:

```
Device(config-ip-sla-dns)# history filter failures
```

(Optional) Defines the type of information kept in the history table for an IP SLAs operation.

Step 9 **frequency** *seconds*

Example:

```
Device(config-ip-sla-dns)# frequency 30
```

(Optional) Sets the rate at which a specified IP SLAs operation repeats.

Step 10 **history hours-of-statistics-kept** *hours*

Example:

```
Device(config-ip-sla-dns)# history hours-of-statistics-kept 4
```

(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.

Step 11 **history lives-kept** *lives*

Example:

```
Device(config-ip-sla-dns)# history lives-kept 5
```

(Optional) Sets the number of lives maintained in the history table for an IP SLAs operation.

Step 12 **owner** *owner-id*

Example:

```
Device(config-ip-sla-dns)# owner admin
```

(Optional) Configures the Simple Network Management Protocol (SNMP) owner of an IP SLAs operation.

Step 13 **history statistics-distribution-interval** *milliseconds*

Example:

```
Device(config-ip-sla-dns)# history statistics-distribution-interval 10
```

(Optional) Sets the time interval for each statistics distribution kept for an IP SLAs operation.

Step 14 **tag** *text*

Example:

```
Device(config-ip-sla-dns)# tag TelnetPollServer1
```

(Optional) Creates a user-specified identifier for an IP SLAs operation.

Step 15 **threshold** *milliseconds*

Example:

```
Device(config-ip-sla-dns)# threshold 10000
```

(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.

Step 16 **timeout** *milliseconds*

Example:

```
Device(config-ip-sla-dns)# timeout 10000
```

(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.

Step 17 **end**

Example:

```
Device(config-ip-sla-dns)# end
```

Exits to privileged EXEC mode.

Schedule IP SLAs operations

Perform this task to schedule IP SLAs operations.

Before you begin

- All IP SLAs operations to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group must be the same.

- The list of one or more operation ID numbers to be added to a multioperation group must be limited to a maximum of 125 characters in length, including commas (,).

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh:mm:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*}] [**ageout** *seconds*] [**recurring**]

Example:

```
Device(config)# ip sla schedule 10 life forever start-time
OR
Device(config)# ip sla schedule 1 3,4,6-9 schedule-period 50 frequency range 80-100
```

(Optional) Configures the scheduling parameters for an individual IP SLAs operation.

- *operation-number*: The IP SLA operation number to schedule (must match a previously created IP SLA operation).

  The range is from 1 to 2147483647.

- **life** {**forever** | *seconds*}: How long the operation will run.
  - **forever**: Runs the operation continuously until manually stopped.
  - *seconds*: Number of seconds the operation should run.

    The range is from 1 to 2147483647 seconds.

- **start-time** {[*hh:mm:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*}: Specifies when to start the operation.
  - *hh:mm:ss* [*month day* | *day month*]: Specific time and date.
  - **pending**: Waits for a manual start.
  - **now**: Starts immediately.
  - **after** *hh:mm:ss*: Starts after the specified amount of time.

- **ageout** *seconds*: Time (in seconds) after which the operation is automatically deleted.

The range is from 0 to 2147483647 seconds.

- **recurring**: Makes the operation run repeatedly according to its frequency settin

Step 4 **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* |**schedule-together**}[**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm* [*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm*[*:ss*]}]

Example:

```
Device(config)# ip sla group schedule 10 schedule-period frequency
OR
Device(config)# ip sla group schedule 1 3,4,6-9 life forever start-time now
```

(Optional) Specifies an IP SLAs operation group number and the range of operation numbers for a multioperation scheduler.

- *group-operation-number*: The number assigned to the group operation (must be unique).

  The range is from 1 to 2147483647.

- *operation-id-numbers*: List of individual IP SLA operation numbers to be included in the group.

  The range is from 1 to 2147483647 (can be a series separated by spaces).

- **schedule-period** *schedule-period-range*: Schedules each operation in the group with a specified time period between them.

  The range is from 1 to 604800 (seconds; up to 7 days).

- **schedule-together**: Starts all operations in the group at the same time.

- **frequency** *group-operation-frequency*: How often (in seconds) the group operation runs.

  The range is from 1 to 604800 seconds.

Step 5 **end**

Example:

```
Device(config)# end
```

Exits global configuration mode and returns to privileged EXEC mode.

Configuration example for a DNS operation

The following example shows how to configure a DNS operation from Device B to the DNS server (IP address 172.20.2.132) as shown in the “DNS Operation” figure in the “DNS Operation” section. The operation is scheduled to start immediately. In this example, the target address is a hostname and the DNS operation will query the DNS server for the IP address associated with the hostname host1. No configuration is required at the DNS server

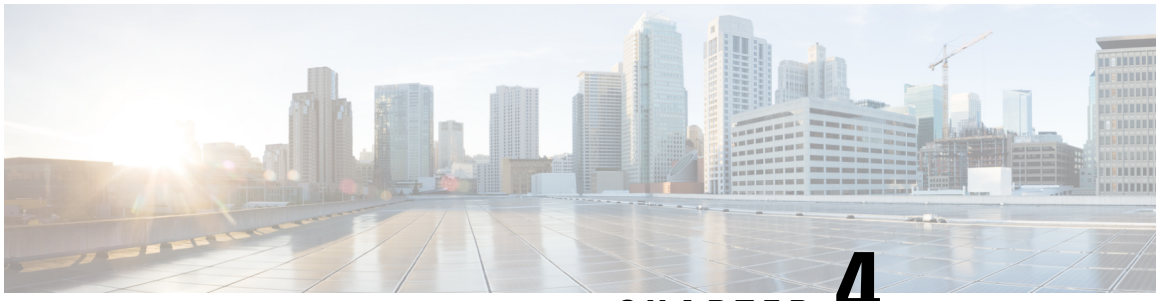
Device B Configuration

```
Device> enable
Device# configure terminal
```

```
Device(config)# ip sla 11
Device(config-ip-sla)# dns host1 name-server 172.20.2.132
Device(config-ip-sla)# frequency 50
Device(config-ip-sla)# timeout 8000
Device(config-ip-sla)# tag DNS-Test
Device(config-ip-sla)# exit
Device(config-ip-sla)# ip sla schedule 11 start-time now
```

Verify IP SLA operations

Command	Description	
show ip sla group schedule	Displays IP SLAs group schedule details.	
show ip sla configuration	Displays IP SLAs configuration details.	



CHAPTER 4

IP SLAs FTP Operation

Feature History for IP SLAs - FTP Operation

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	IP SLAs - FTP Operation: This operation measures the response time between a Cisco device and an FTP server when retrieving a file.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

IP SLAs FTP

The IP SLAs FTP operation is designed to measure the response time between a Cisco device and an File Transfer Protocol (FTP) server when retrieving a file. This operation supports only FTP GET requests, allowing administrators to assess the performance of FTP file transfers across the network. By displaying and analyzing the results of the FTP operation, network capacity and FTP server performance can be evaluated, making it a valuable tool for both ongoing monitoring and troubleshooting of FTP-related issues.

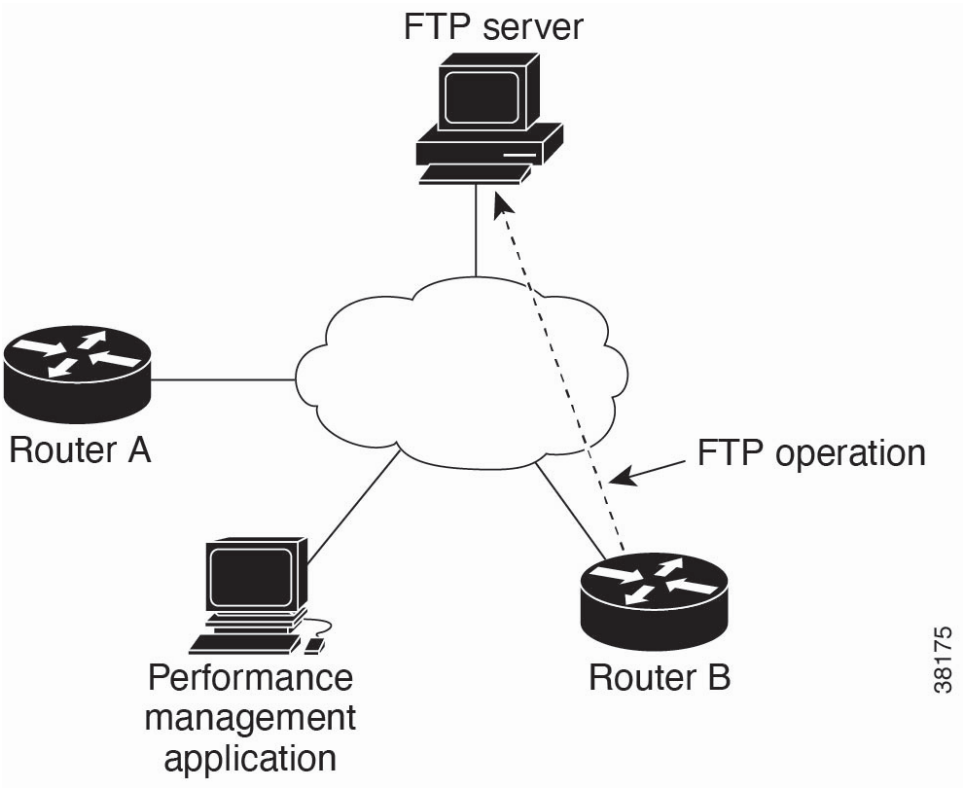
The FTP operation also measures your FTP server performance levels by determining the RTT taken to retrieve a file.

How IP SLAs FTP works

The IP SLAs FTP operation measures the round-trip time (RTT) between a Cisco device and an FTP server when retrieving a file. As an application protocol within the TCP/IP protocol stack, FTP is commonly used for transferring files between devices on a network. By performing an FTP GET request, the operation provides insight into the performance and responsiveness of file transfers, enabling network administrators to monitor and troubleshoot FTP connectivity and server performance effectively.

In the figure below Device B is configured as the source IP SLAs device and an FTP operation is configured with the FTP server as the destination device.

Figure 5: FTP Operation



Connection response time is computed by measuring the time taken to download a file to Device B from the remote FTP server using FTP over TCP. This operation does not use the IP SLAs Responder.

Note To test the response time to connect to an FTP port (Port 21), use the IP SLAs TCP Connect operation.

FTP transfer modes

Both active and passive FTP transfer modes are supported. The passive mode is enabled by default.

FTP operation type

Only the FTP GET (download) operation type is supported. The URL specified for the FTP GET operation must be in one of the following formats:

- ftp://username:password@host/filename
- ftp://host/filename

If the username and password are not specified, the defaults are anonymous and test, respectively.

FTP and network performance

FTP carries a significant amount of data traffic and can affect the performance of your network. The results of an IP SLAs FTP operation to retrieve a large file can be used to determine the capacity of the network but retrieve large files with caution because the FTP operation will consume more bandwidth.

Guidelines to configure IP SLAs FTP

The IP SLAs FTP operation only supports FTP GET (download) requests

Configure an IP SLAs FTP operation on the source device

Follow the steps in each of these tasks to configure an IP SLAs FTP operation on the source device.

Before you begin

There is no need to configure an IP SLAs responder on the destination device.

Procedure

Step 1 Perform any one of these tasks:

- Configure a Basic FTP operation on a source device
- Configure an FTP operation with optional parameters on the source device

Step 2 Schedule IP SLAs operations

Configure a Basic FTP operation on a source device

Perform this task to configure a basic FTP operation on a source device.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla** *operation-number*

Example:

```
Device(config)# ip sla 10
```

Starts configuring an IP SLAs operation and enters IP SLA configuration mode.

Step 4 **ftp get** *url* [**source-ip** {*ip-address* | *hostname*}] [**mode** {**passive** | **active**}

Example:

```
Device(config-ip-sla)# ftp get ftp://username:password@hostip/test.cap
```

Defines an FTP operation and enters IP SLA FTP configuration mode.

- *url*: The FTP URL of the file to retrieve. Acceptable formats are:
  - ftp://username:password@host/filename
  - ftp://host/filename
- **source-ip** {*ip-address* | *hostname*}: (Optional) Specifies the source IP address or hostname to use for the FTP request.
- **mode** {**passive** | **active**}: (Optional) Sets the FTP transfer mode. By default, passive mode is enabled.

Step 5 **frequency** *seconds*

Example:

```
Device(config-ip-sla-ftp)# frequency 90
```

(Optional) Sets the rate at which a specified IP SLAs operation repeats.

Step 6 **end**

Example:

```
Device(config-ip-sla-ftp)# end
```

Exits to privileged EXEC mode.

Configure an FTP operation with optional parameters on the source device

Perform this task to configure an FTP operation with optional parameters on the source device.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla** *operation-number*

Example:

```
Device(config)# ip sla 10
```

Starts configuring an IP SLAs operation and enters IP SLA configuration mode.

Step 4 **ftp get** *url* [**source-ip** {*ip-address* | *hostname*}] [**mode** {**passive** | **active**}

Example:

```
Device(config-ip-sla)# ftp get ftp://username:password@hostip/test.cap
```

Defines an FTP operation and enters IP SLA FTP configuration mode.

- *url*: The FTP URL of the file to retrieve. Acceptable formats are:
  - ftp://username:password@host/filename
  - ftp://host/filename
- **source-ip** {*ip-address* | *hostname*}: (Optional) Specifies the source IP address or hostname to use for the FTP request.
- **mode** {**passive** | **active**}: (Optional) Sets the FTP transfer mode. By default, passive mode is enabled.

Step 5 **history buckets-kept** *size*

Example:

```
Device(config-ip-sla-ftp)# history buckets-kept 25
```

(Optional) Sets the number of history buckets that are kept during the lifetime of an IP SLAs operation.

Step 6 **history distributions-of-statistics-kept** *size*

Example:

```
Device(config-ip-sla-ftp)# history distributions-of-statistics-kept 5
```

(Optional) Sets the number of statistics distributions kept per hop during an IP SLAs operation.

Step 7 **history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*]

Example:

```
Device(config-ip-sla-ftp)# history enhanced interval 900 buckets 100
```

(Optional) Enables enhanced history gathering for an IP SLAs operation.

- **interval** *seconds*: (Optional) The time interval, in seconds, between each statistics recording.
- **buckets** *number-of-buckets*: (Optional) The number of history data buckets (records) to store for the operation.

Step 8 **history filter** {**none** | **all** | **overThreshold** | **failures**}

Example:

```
Device(config-ip-sla-ftp)# history filter failures
```

(Optional) Defines the type of information kept in the history table for an IP SLAs operation.

Step 9 **frequency** *seconds*

Example:

```
Device(config-ip-sla-ftp)# frequency 30
```

(Optional) Sets the rate at which a specified IP SLAs operation repeats.

Step 10 **history hours-of-statistics-kept** *hours*

Example:

```
Device(config-ip-sla-ftp)# history hours-of-statistics-kept 4
```

(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.

Step 11 **history lives-kept** *lives*

Example:

```
Device(config-ip-sla-ftp)# history lives-kept 5
```

(Optional) Sets the number of lives maintained in the history table for an IP SLAs operation.

Step 12 **owner** *owner-id*

Example:

```
Device(config-ip-sla-ftp)# owner admin
```

(Optional) Configures the Simple Network Management Protocol (SNMP) owner of an IP SLAs operation.

Step 13 **history statistics-distribution-interval** *milliseconds*

Example:

```
Device(config-ip-sla-ftp)# history statistics-distribution-interval 10
```

(Optional) Sets the time interval for each statistics distribution kept for an IP SLAs operation.

Step 14 **tag** *text*

Example:

```
Device(config-ip-sla-ftp)# tag TelnetPollServer1
```

(Optional) Creates a user-specified identifier for an IP SLAs operation.

Step 15 **threshold** *milliseconds*

Example:

```
Device(config-ip-sla-ftp)# threshold 10000
```

(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.

Step 16 **timeout** *milliseconds*

Example:

```
Device(config-ip-sla-ftp)# timeout 10000
```

(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.

Step 17 **end**

Example:

```
Device(config-ip-sla-ftp)# end
```

Exits to privileged EXEC mode.

Schedule IP SLAs operations

Perform this task to schedule IP SLAs operations.

Before you begin

- All IP SLAs operations to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group must be the same.
- The list of one or more operation ID numbers to be added to a multioperation group must be limited to a maximum of 125 characters in length, including commas (,).

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh:mm:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*}] [**ageout** *seconds*] [**recurring**]

Example:

```
Device(config)# ip sla schedule 10 life forever start-time
OR
Device(config)# ip sla schedule 1 3,4,6-9 schedule-period 50 frequency range 80-100
```

(Optional) Configures the scheduling parameters for an individual IP SLAs operation.

- *operation-number*: The IP SLA operation number to schedule (must match a previously created IP SLA operation).

  The range is from 1 to 2147483647.

- **life** {**forever** | *seconds*}: How long the operation will run.
  - **forever**: Runs the operation continuously until manually stopped.
  - *seconds*: Number of seconds the operation should run.

    The range is from 1 to 2147483647 seconds.

- **start-time** {[*hh:mm:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*}: Specifies when to start the operation.
  - *hh:mm:ss* [*month day* | *day month*]: Specific time and date.
  - **pending**: Waits for a manual start.
  - **now**: Starts immediately.
  - **after** *hh:mm:ss*: Starts after the specified amount of time.

- **ageout** *seconds*: Time (in seconds) after which the operation is automatically deleted.

  The range is from 0 to 2147483647 seconds.

- **recurring**: Makes the operation run repeatedly according to its frequency settin

Step 4 **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* |**schedule-together**}[**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm* [*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm*[*:ss*]}]

Example:

```
Device(config)# ip sla group schedule 10 schedule-period frequency
OR
Device(config)# ip sla group schedule 1 3,4,6-9 life forever start-time now
```

(Optional) Specifies an IP SLAs operation group number and the range of operation numbers for a multioperation scheduler.

- *group-operation-number*: The number assigned to the group operation (must be unique).

  The range is from 1 to 2147483647.

- *operation-id-numbers*: List of individual IP SLA operation numbers to be included in the group.

  The range is from 1 to 2147483647 (can be a series separated by spaces).

- **schedule-period** *schedule-period-range*: Schedules each operation in the group with a specified time period between them.

  The range is from 1 to 604800 (seconds; up to 7 days).

- **schedule-together**: Starts all operations in the group at the same time.

- **frequency** *group-operation-frequency*: How often (in seconds) the group operation runs.

  The range is from 1 to 604800 seconds.

Step 5 **end**

Example:

```
Device(config)# end
```

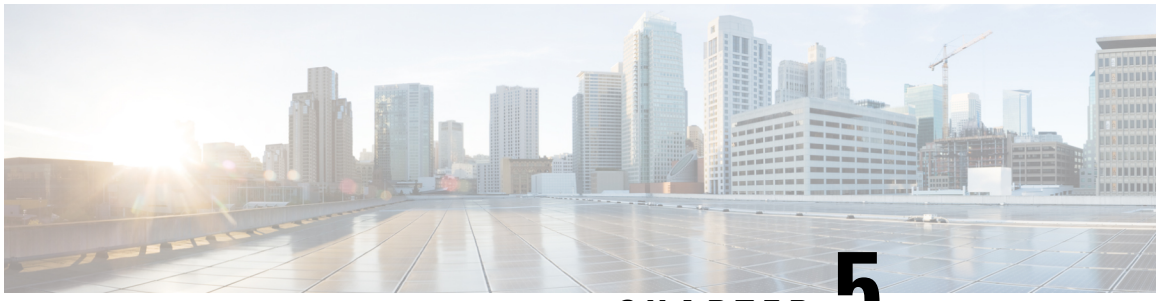
Exits global configuration mode and returns to privileged EXEC mode.

Configuration example for an FTP operation

The following example shows how to configure an FTP operation from Device B to the FTP server as shown in the "FTP Operation" figure in the "Information About IP SLAs FTP Operation" section. The operation is scheduled to start every day at 1:30 a.m. In this example, the file named test.cap is to be retrieved from the host, cisco.com, with a password of abc using FTP in active mode.

Device B Configuration

```
Device> enable
Device# configure terminal
Device(config)# ip sla 10
Device(config-ip-sla)# ftp get ftp://user1:abc@test.cisco.com/test.cap mode active
Device(config-ip-sla)# frequency 20
Device(config-ip-sla)# tos 128
Device(config-ip-sla)# timeout 40000
Device(config-ip-sla)# tag FLL-FTP
Device(config-ip-sla)# ip sla schedule 10 start-time 01:30:00 recurring
```



CHAPTER 5

IP SLAs HTTP Operation

Feature History for IP SLAs - HTTP Operation

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	IP SLAs - HTTP Operation: This operation monitors the response time between a Cisco device and an HTTP server when retrieving a web page.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

IP SLAs HTTP

The IP SLAs HTTP operation is used to monitor the response time between a Cisco device and an HTTP server when retrieving a web page. This operation supports both standard GET requests and custom RAW requests, providing flexibility in how HTTP traffic is tested. By analyzing the results of the HTTP operation, network administrators can assess the performance of an HTTP server, identify potential issues, and ensure reliable web service delivery. The collected data can be displayed and reviewed to gain insights into server responsiveness and overall network health.

How IP SLAs HTTP work

The IP SLA HTTP operation measures the response time required to retrieve a web page from an HTTP server, providing valuable insight into web server performance. This operation involves three key response time measurements:

- DNS lookup RTT (the round-trip time to resolve the domain name),
- TCP Connect RTT (the time to establish a TCP connection to the server), and
- HTTP transaction RTT (the time taken to send an HTTP request and receive the response, specifically for the home HTML page)

Additionally, the "time to first byte" metric records the duration from the start of the TCP connection to the receipt of the first HTML byte. The total HTTP RTT is the sum of DNS, TCP Connect, and HTTP transaction times. For GET requests, IP SLAs automatically format the request based on the provided URL, while RAW requests allow complete customization of the HTTP request, enabling control over fields such as authentication. HTTP requests can also be routed through a proxy server. The results of these operations help monitor web server performance by measuring the round-trip time to retrieve a web page. The operation continues to function regardless of HTTP errors, but the IP SLA HTTP operation is considered down only if the returned HTTP code is not 200.

Note The only time the SLA probe goes down is when the SLA is unable to establish a TCP connection or is unable to receive an answer from the Remote server to its HTTP request.

Guidelines to configure IP SLAs HTTP operations

- IP SLAs HTTP operations support only HTTP/1.0.
- HTTP/1.1 is not supported for any IP SLAs HTTP operation, including HTTP RAW requests.

Configure an HTTP GET operation on the source device

Follow the steps in each of these tasks to configure an HTTP GET operation on the source device.

Procedure

Step 1 Perform any one of these tasks:

- Configure a basic HTTP GET operation on the source device
- Configure an HTTP GET operation with optional parameters on the source device
- Configure an HTTP RAW operation on the source device

Step 2 Schedule IP SLAs operations

Configure a basic HTTP GET operation on the source device

Perform this task to configure a basic HTTP GET operation on the source device.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla** *operation-number*

Example:

```
Device(config)# ip sla 10
```

Starts configuring an IP SLAs operation and enters IP SLA configuration mode.

Step 4 **http** {**get** | **raw**} *url* [**name-server** *ip-address*] [**version** *version-number*] [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**cache** {**enable** | **disable**}] [**proxy** *proxy-url*]

Example:

```
Device(config-ip-sla)# http get http://198.133.219.25
```

Defines an HTTP operation and enters IP SLA configuration mode.

- **get**: Specifies a standard HTTP GET request.
- **raw**: Allows you to send a custom RAW HTTP request.
- *url*: The target URL to retrieve.
- **name-server** *ip-address*: (Optional) Specifies the DNS server to use for name resolution.
- **version** *version-number*: (Optional) Sets the HTTP version to use for example 1.0, 1.1.
- **source-ip** {*ip-address* | *hostname*}: (Optional) Specifies the source IP address or hostname to use for the request.
- **source-port** *port-number*: (Optional) Specifies the source port number for the request.
- **cache** {**enable** | **disable**}: (Optional) Enables or disables caching for the request.
- **proxy** *proxy-url*: (Optional) Specifies the proxy server to use for the HTTP request.

Step 5 **frequency** *seconds*

Example:

```
Device(config-ip-sla-http)# frequency 90
```

(Optional) Sets the rate at which a specified IP SLAs HTTP operation repeats. The default and minimum frequency value for an IP SLAs HTTP operation is 60 seconds.

Step 6 **end**

Example:

```
Device(config-ip-sla-http)# end
```

Exits to privileged EXEC mode.

Configure an HTTP GET operation with optional parameters on the source device

Perform this task to configure an HTTP GET operation with optional parameters on the source device.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla** *operation-number*

Example:

```
Device(config)# ip sla 10
```

Starts configuring an IP SLAs operation and enters IP SLA configuration mode.

Step 4 **http** {**get** | **raw**} *url* [**name-server** *ip-address*] [**version** *version-number*] [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**cache** {**enable** | **disable**}] [**proxy** *proxy-url*]

Example:

```
Device(config-ip-sla)# http get http://198.133.219.25
```

Defines an HTTP operation and enters IP SLA configuration mode.

- **get**: Specifies a standard HTTP GET request.

- **raw**: Allows you to send a custom RAW HTTP request.
- *url*: The target URL to retrieve.
- **name-server** *ip-address*: (Optional) Specifies the DNS server to use for name resolution.
- **version** *version-number*: (Optional) Sets the HTTP version to use for example 1.0, 1.1.
- **source-ip** {*ip-address* | *hostname*}: (Optional) Specifies the source IP address or hostname to use for the request.
- **source-port** *port-number*: (Optional) Specifies the source port number for the request.
- **cache** {**enable** | **disable**}: (Optional) Enables or disables caching for the request.
- **proxy** *proxy-url*: (Optional) Specifies the proxy server to use for the HTTP request.

Step 5 **history distributions-of-statistics-kept** *size*

Example:

```
Device(config-ip-sla-http)# history distributions-of-statistics-kept 5
```

(Optional) Sets the number of statistics distributions kept per hop during an IP SLAs operation.

Step 6 **frequency** *seconds*

Example:

```
Device(config-ip-sla-http)# frequency 90
```

(Optional) Sets the rate at which a specified IP SLAs HTTP operation repeats. The default and minimum frequency value for an IP SLAs HTTP operation is 60 seconds.

Step 7 **history hours-of-statistics-kept** *hours*

Example:

```
Device(config-ip-sla-http)# history hours-of-statistics-kept 4
```

(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.

Step 8 **http-raw-request**

Example:

```
Device(config-ip-sla-http)# http-raw-request
```

(Optional) Explicitly specifies the options for a GET request for an IP SLAs HTTP operation.

Step 9 **owner** *owner-id*

Example:

```
Device(config-ip-sla-http)# owner admin
```

(Optional) Configures the Simple Network Management Protocol (SNMP) owner of an IP SLAs operation.

Step 10 **history statistics-distribution-interval** *milliseconds*

Example:

```
Device(config-ip-sla-http)# history statistics-distribution-interval 10
```

(Optional) Sets the time interval for each statistics distribution kept for an IP SLAs operation.

Step 11 **tag** *text*

Example:

```
Device(config-ip-sla-http)# tag TelnetPollServer1
```

(Optional) Creates a user-specified identifier for an IP SLAs operation.

Step 12 **threshold** *milliseconds*

Example:

```
Device(config-ip-sla-http)# threshold 10000
```

(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.

Step 13 **timeout** *milliseconds*

Example:

```
Device(config-ip-sla-http)# timeout 10000
```

(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.

Step 14 **tos** *number*

Example:

```
Device(config-ip-sla-http)# tos 160
```

(Optional) Defines a type of service (ToS) byte in the IP header of an IP SLAs operation.

Step 15 **end**

Example:

```
evice(config-ip-sla-http)# end
```

Exits to privileged EXEC mode.

Configure an HTTP RAW operation on the source device

Before you begin

This operation does not require an IP SLAs Responder on the destination device.

Perform this task to configure an HTTP RAW operation on the source device.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla** *operation-number*

Example:

```
Device(config)# ip sla 10
```

Starts configuring an IP SLAs operation and enters IP SLA configuration mode.

Step 4 **http** {**get** | **raw**} *url* [**name-server** *ip-address*] [**version** *version-number*] [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**cache** {**enable** | **disable**}] [**proxy** *proxy-url*]

Example:

```
Device(config-ip-sla)# http get http://198.133.219.25
```

Defines an HTTP operation and enters IP SLA configuration mode.

- **get**: Specifies a standard HTTP GET request.
- **raw**: Allows you to send a custom RAW HTTP request.
- *url*: The target URL to retrieve.
- **name-server** *ip-address*: (Optional) Specifies the DNS server to use for name resolution.
- **version** *version-number*: (Optional) Sets the HTTP version to use for example 1.0, 1.1.
- **source-ip** {*ip-address* | *hostname*}: (Optional) Specifies the source IP address or hostname to use for the request.
- **source-port** *port-number*: (Optional) Specifies the source port number for the request.
- **cache** {**enable** | **disable**}: (Optional) Enables or disables caching for the request.
- **proxy** *proxy-url*: (Optional) Specifies the proxy server to use for the HTTP request.

Step 5 **http-raw-request**

Example:

```
Device(config-ip-sla)# http-raw-request
```

Enters HTTP RAW configuration mode.

Step 6 Enter the required HTTP 1.0 command syntax.

Example:

```
Device(config-ip-sla-http)# GET /en/US/hmpgs/index.html HTTP/1.0\r\n\r\n
```

Specifies all the required HTTP 1.0 commands.

Step 7 **end**

Example:

```
Device(config-ip-sla-http)# end
```

Exits to privileged EXEC mode.

Schedule IP SLAs operations

Perform this task to schedule IP SLAs operations.

Before you begin

- All IP SLAs operations to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group must be the same.
- The list of one or more operation ID numbers to be added to a multioperation group must be limited to a maximum of 125 characters in length, including commas (,).

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh:mm:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*}] [**ageout** *seconds*] [**recurring**]

Example:

```
Device(config)# ip sla schedule 10 life forever start-time
OR
Device(config)# ip sla schedule 1 3,4,6-9 schedule-period 50 frequency range 80-100
```

(Optional) Configures the scheduling parameters for an individual IP SLAs operation.

- *operation-number*: The IP SLA operation number to schedule (must match a previously created IP SLA operation).

  The range is from 1 to 2147483647.

- **life** {**forever** | *seconds*}: How long the operation will run.
  - **forever**: Runs the operation continuously until manually stopped.
  - *seconds*: Number of seconds the operation should run.

    The range is from 1 to 2147483647 seconds.

- **start-time** {[*hh:mm:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*}: Specifies when to start the operation.
  - *hh:mm:ss* [*month day* | *day month*]: Specific time and date.
  - **pending**: Waits for a manual start.
  - **now**: Starts immediately.
  - **after** *hh:mm:ss*: Starts after the specified amount of time.
- **ageout** *seconds*: Time (in seconds) after which the operation is automatically deleted.

  The range is from 0 to 2147483647 seconds.
- **recurring**: Makes the operation run repeatedly according to its frequency settin

Step 4 **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* |**schedule-together**}[**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm* [*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm*[*:ss*]}]

Example:

```
Device(config)# ip sla group schedule 10 schedule-period frequency
OR
Device(config)# ip sla group schedule 1 3,4,6-9 life forever start-time now
```

(Optional) Specifies an IP SLAs operation group number and the range of operation numbers for a multioperation scheduler.

- *group-operation-number*: The number assigned to the group operation (must be unique).

  The range is from 1 to 2147483647.
- *operation-id-numbers*: List of individual IP SLA operation numbers to be included in the group.

  The range is from 1 to 2147483647 (can be a series separated by spaces).
- **schedule-period** *schedule-period-range*: Schedules each operation in the group with a specified time period between them.

  The range is from 1 to 604800 (seconds; up to 7 days).
- **schedule-together**: Starts all operations in the group at the same time.
- **frequency** *group-operation-frequency*: How often (in seconds) the group operation runs.

  The range is from 1 to 604800 seconds.

Step 5 **end**

Example:

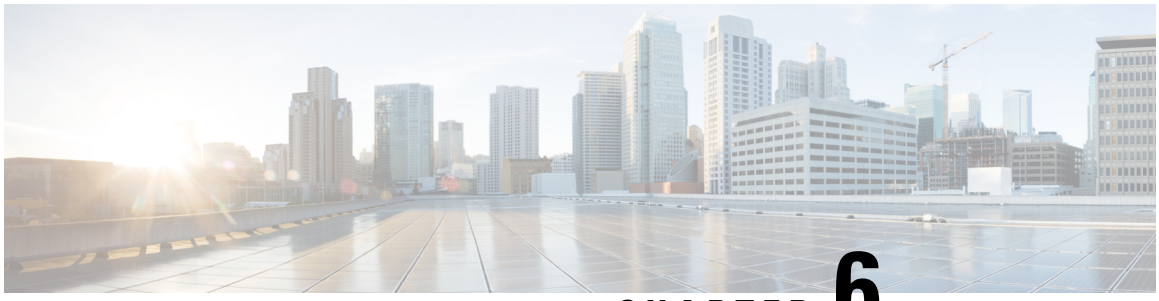
```
Device(config)# end
```

Exits global configuration mode and returns to privileged EXEC mode.

Configuration example for an HTTP RAW operation with authentication

The following example shows how to configure an HTTP RAW operation with authentication.

```
Device> enable
Device# configure terminal
Device(config)# ip sla 8
Device(config-ip-sla)# http raw url http://site-test.cisco.com
Device(config-ip-sla)# http-raw-request
Device(config-ip-sla)# GET /lab/index.html HTTP/1.0\r\n
Authorization: Basic btNpdGT4biNvoZe=\r\n
\r\n
Device(config-ip-sla)# end
Device# configure terminal
Device(config)# ip sla schedule 8 life forever start-time now
```



CHAPTER 6

IP SLAs ICMP Path Echo Operation

Feature History for IP SLAs - ICMP Path Echo Operation

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	IP SLAs - ICMP Path Echo Operation: This operation monitors both end-to-end and hop-by-hop response times between a Cisco device and other IP-enabled devices.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

IP SLAs ICMP Path Echo

IP SLAs ICMP Path Echo operation allows for monitoring both end-to-end and hop-by-hop response times between a Cisco device and other IP-enabled devices. By sending Internet Control Message Protocol (ICMP) packets along the network path, this operation helps network administrators assess network availability and efficiently troubleshoot connectivity issues. The collected results provide insights into the performance of ICMP across the network, allowing for detailed analysis of response times at each hop and enabling proactive identification and resolution of potential network problems.

How IP SLAs ICMP path echo works

To monitor ICMP Path Echo performance on a device, the IP SLAs ICMP Path Echo operation can be utilized. This operation measures both end-to-end and hop-by-hop response times between a Cisco device and other IP-enabled devices. By providing detailed response time data, ICMP Path Echo helps network administrators determine network availability and effectively troubleshoot network connectivity issues, ensuring optimal network performance and reliability.

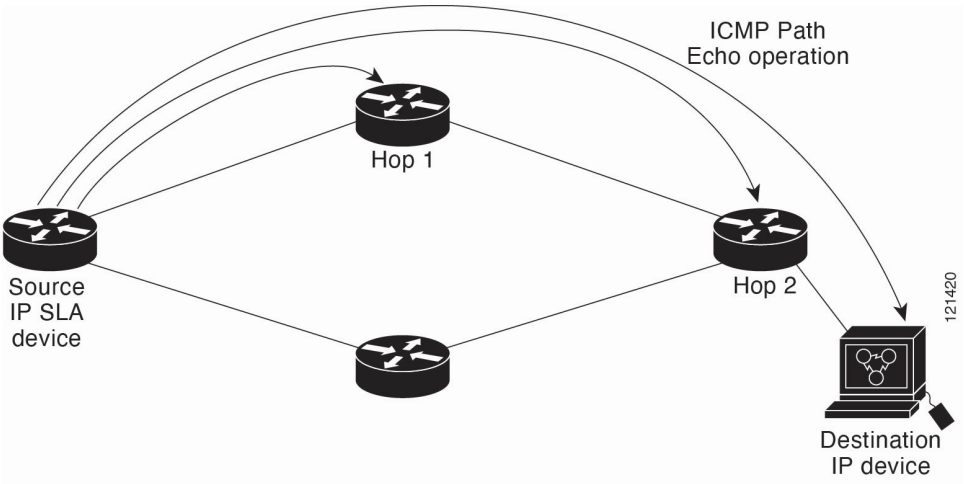
Summary

The IP SLAs ICMP Path Echo operation records detailed statistics for each hop along the network path taken to reach its destination. By leveraging the traceroute facility, this operation identifies each hop and measures the response time between a Cisco device and any IP device on the network. This hop-by-hop analysis enables precise monitoring and troubleshooting, allowing network administrators to pinpoint delays or issues at specific points along the path.

In the figure below the source IP SLAs device uses traceroute to discover the path to the destination IP device. A ping is then used to measure the response time between the source IP SLAs device and each subsequent hop in the path to the destination IP device.

Workflow

Figure 6: ICMP Path Echo Operation



Using the statistics recorded for the response times and availability, the ICMP Path Echo operation can identify a hop in the path that is causing a bottleneck.

Guidelines to configure IP SLAs ICMP Path Echo Operations

We recommend using a Cisco networking device as the destination device although any networking device that supports RFC 862, Echo protocol, can be used.

Configure an IP SLAs ICMP Path Echo operations on the source device

Follow the steps in each of these tasks to configure an ICMP Path Echo operations on the source device.

Before you begin

This operation does not require an IP SLAs Responder on the destination device.

Procedure

Step 1 Perform any one of these tasks:

- Configure a basic ICMP path echo operation on the source device
- Configure an ICMP path echo operation with optional parameters on the source device

Step 2 Schedule IP SLAs operations

Configure a basic ICMP path echo operation on the source device

Before you begin

This operation does not require an IP SLAs Responder on the destination device.

Perform this task to configure a basic ICMP path echo operation on the source device.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla** *operation-number*

Example:

```
Device(config)# ip sla 10
```

Starts configuring an IP SLAs operation and enters IP SLA configuration mode.

Step 4 **path-echo** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*}]

Example:

```
Device(config-ip-sla)# path-echo 172.29.139.134
```

Defines a path echo operation and enters IP SLA path echo configuration mode.

- *destination-ip-address*: The IP address of the target device to which the path echo operation will be sent.
- *destination-hostname*: The hostname of the target device (as an alternative to specifying the IP address).
- **source-ip** {*ip-address* | *hostname*}: (Optional) Specifies the source IP address or hostname from which the operation is initiated.

Step 5 **frequency** *seconds*

Example:

```
Device(config-ip-sla-pathEcho)# frequency 30
```

(Optional) Sets the rate at which a specified IP SLAs operation repeats.

Step 6 **end**

Example:

```
Device(config-ip-sla-pathEcho)# end
```

Exits to privileged EXEC mode.

Configure an ICMP path echo operation with optional parameters on the source device

Perform this task to configure an ICMP path echo operation with optional parameters on the source device.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla** *operation-number*

Example:

```
Device(config)# ip sla 10
```

Starts configuring an IP SLAs operation and enters IP SLA configuration mode.

Step 4 **path-echo** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*}]

Example:

```
Device(config-ip-sla)# path-echo 172.29.139.134
```

Defines a path echo operation and enters IP SLA path echo configuration mode.

- *destination-ip-address*: The IP address of the target device to which the path echo operation will be sent.
- *destination-hostname*: The hostname of the target device (as an alternative to specifying the IP address).
- **source-ip** {*ip-address* | *hostname*}: (Optional) Specifies the source IP address or hostname from which the operation is initiated.

Step 5 **history buckets-kept** *size*

Example:

```
Device(config-ip-sla-pathEcho)# history buckets-kept 25
```

(Optional) Sets the number of history buckets that are kept during the lifetime of an IP SLAs operation.

Step 6 **history distributions-of-statistics-kept** *size*

Example:

```
Device(config-ip-sla-pathEcho)# history distributions-of-statistics-kept 5
```

(Optional) Sets the number of statistics distributions kept per hop during an IP SLAs operation.

Step 7 **history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*]

Example:

```
Device(config-ip-sla-pathEcho)# history enhanced interval 900 buckets 100
```

(Optional) Enables enhanced history gathering for an IP SLAs operation.

- **interval** *seconds*: (Optional) The time interval, in seconds, at which statistics are recorded.
- **buckets** *number-of-buckets*: (Optional) The number of history buckets (data storage slots) to retain for the operation.

Step 8 **history filter** {**none** | **all** | **overThreshold** | **failures**}

Example:

```
Device(config-ip-sla-pathEcho)# history filter failures
```

(Optional) Defines the type of information kept in the history table for an IP SLAs operation.

Step 9 **frequency** *seconds*

Example:

```
Device(config-ip-sla-pathEcho)# frequency 30
```

(Optional) Sets the rate at which a specified IP SLAs operation repeats.

Step 10 **history hours-of-statistics-kept** *hours*

Example:

```
Device(config-ip-sla-pathEcho)# history hours-of-statistics-kept 4
```

(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.

Step 11 **history lives-kept** *lives*

Example:

```
Device(config-ip-sla-pathEcho)# history lives-kept 5
```

(Optional) Sets the number of lives maintained in the history table for an IP SLAs operation.

Step 12 **owner** *owner-id*

Example:

```
Device(config-ip-sla-pathEcho)# owner admin
```

(Optional) Configures the Simple Network Management Protocol (SNMP) owner of an IP SLAs operation.

Step 13 **paths-of-statistics-kept** *size*

Example:

```
ice(config-ip-sla-pathEcho)# paths-of-statistics-kept 3
```

(Optional) Sets the number of paths for which statistics are maintained per hour for an IP SLAs operation.

Step 14 **request-data-size** *bytes*

Example:

```
Device(config-ip-sla-pathEcho)# request-data-size 64
```

(Optional) Sets the protocol data size in the payload of an IP SLAs operation's request packet.

Step 15 **samples-of-history-kept** *samples*

Example:

```
Device(config-ip-sla-pathEcho)# samples-of-history-kept 10
```

(Optional) Sets the number of entries kept in the history table per bucket for an IP SLAs operation.

Step 16 **history statistics-distribution-interval** *milliseconds*

Example:

```
Device(config-ip-sla-pathEcho)# history statistics-distribution-interval 10
```

(Optional) Sets the time interval for each statistics distribution kept for an IP SLAs operation.

Step 17 **tag** *text*

Example:

```
Device(config-ip-sla-pathEcho)# tag TelnetPollServer1
```

(Optional) Creates a user-specified identifier for an IP SLAs operation.

Step 18 **threshold** *milliseconds*

Example:

```
Device(config-ip-sla-pathEcho)# threshold 10000
```

(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.

Step 19 **tos** *number*

Example:

```
Device(config-ip-sla-pathEcho)# tos 160
```

(Optional) Defines a type of service (ToS) byte in the IP header of an IP SLAs operation.

Step 20 **verify-data**

Example:

```
Device(config-ip-sla-pathEcho)# verify-data
```

(Optional) Causes an IP SLAs operation to check each reply packet for data corruption.

Step 21 **vrf** *vrf-name*

Example:

```
Device(config-ip-sla-pathEcho)# vrf vpn-A
```

(Optional) Allows monitoring within Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs) using IP SLAs operations.

Step 22 **end**

Example:

```
Device(config-ip-sla-pathEcho)# end
```

Exits to privileged EXEC mode.

Schedule IP SLAs operations

Perform this task to schedule IP SLAs operations.

Before you begin

- All IP SLAs operations to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group must be the same.
- The list of one or more operation ID numbers to be added to a multioperation group must be limited to a maximum of 125 characters in length, including commas (,).

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh:mm:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*}] [**ageout** *seconds*] [**recurring**]

Example:

```
Device(config)# ip sla schedule 10 life forever start-time
OR
Device(config)# ip sla schedule 1 3,4,6-9 schedule-period 50 frequency range 80-100
```

(Optional) Configures the scheduling parameters for an individual IP SLAs operation.

- *operation-number*: The IP SLA operation number to schedule (must match a previously created IP SLA operation).

  The range is from 1 to 2147483647.

- **life** {**forever** | *seconds*}: How long the operation will run.
  - **forever**: Runs the operation continuously until manually stopped.
  - *seconds*: Number of seconds the operation should run.

    The range is from 1 to 2147483647 seconds.

- **start-time** {[*hh:mm:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*}: Specifies when to start the operation.
  - *hh:mm:ss* [*month day* | *day month*]: Specific time and date.
  - **pending**: Waits for a manual start.
  - **now**: Starts immediately.
  - **after** *hh:mm:ss*: Starts after the specified amount of time.

- **ageout** *seconds*: Time (in seconds) after which the operation is automatically deleted.

  The range is from 0 to 2147483647 seconds.

- **recurring**: Makes the operation run repeatedly according to its frequency settin

Step 4 **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* |**schedule-together**}[**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm* [*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm*[*:ss*]}]

Example:

```
Device(config)# ip sla group schedule 10 schedule-period frequency
OR
Device(config)# ip sla group schedule 1 3,4,6-9 life forever start-time now
```

(Optional) Specifies an IP SLAs operation group number and the range of operation numbers for a multioperation scheduler.

- *group-operation-number*: The number assigned to the group operation (must be unique).

  The range is from 1 to 2147483647.

- *operation-id-numbers*: List of individual IP SLA operation numbers to be included in the group.

  The range is from 1 to 2147483647 (can be a series separated by spaces).

- **schedule-period** *schedule-period-range*: Schedules each operation in the group with a specified time period between them.

  The range is from 1 to 604800 (seconds; up to 7 days).

- **schedule-together**: Starts all operations in the group at the same time.

- **frequency** *group-operation-frequency*: How often (in seconds) the group operation runs.

  The range is from 1 to 604800 seconds.

Step 5 **end**

Example:

```
Device(config)# end
```

Exits global configuration mode and returns to privileged EXEC mode.

Configuration examples for IP SLAs ICMP Path Echo operations

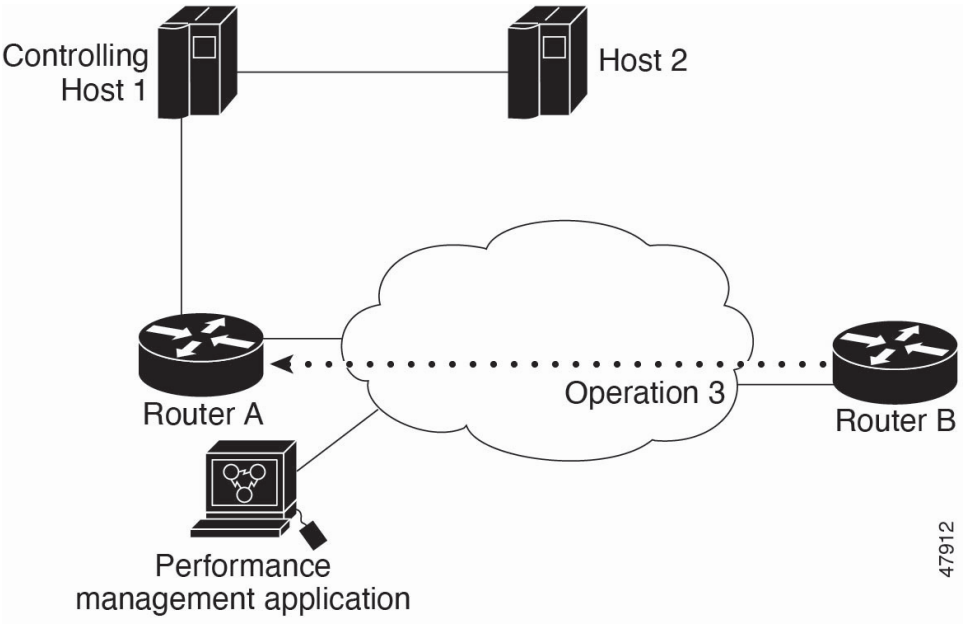
These sections provide configuration examples for IP SLAs ICMP Path Echo operation.

Example: Configure an ICMP path echo operation

The following example shows how to configure an IP SLAs operation type of ICMP Path Echo that will start after 30 seconds and run for 5 minutes.

The figure below depicts the ICMP Path Echo operation.

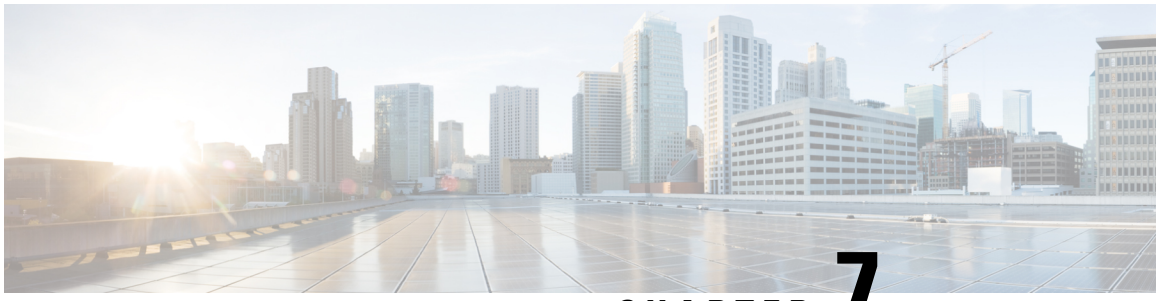
Figure 7: ICMP Path Echo Operation



This example sets a Path Echo operation (ip sla 3) from Device B to Device A using IP/ICMP. The operation attempts to execute three times in 25 seconds (first attempt at 0 seconds).

Device B Configuration

```
Device> enable
Device# configure terminal
Device(config)# ip sla 3
Device(config-ip-sla)# path-echo 172.29.139.134
Device(config-ip-sla-pathEcho)# frequency 10
Device(config-ip-sla-pathEcho)# tag SGN-RO
Device(config-ip-sla-pathEcho)# timeout 1000
Device(config-ip-sla-pathEcho)# ip sla schedule 3 life 25
Device(config-ip-sla-pathEcho)# end
```



CHAPTER 7

IP SLAs LSP Health Monitor

Feature History for IP SLAs - LSP Health Monitor

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	IP SLAs - LSP Health Monitor: This feature enables proactive monitoring of Layer 3 MPLS VPNs by providing automated end-to-end verification in both the control plane and data plane for all LSPs between participating PE devices.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

IP SLAs LSP Health Monitor Operations

LSP Health Monitors enable proactive monitoring of Layer 3 Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs) by providing automated end-to-end verification in both the control plane and data plane for all Label Switched Paths (LSPs) between participating Provider Edge (PE) devices. This PE-to-PE device approach ensures that LSP connectivity is verified along the actual paths used by customer traffic, allowing detection of customer-impacting network connectivity issues within the MPLS core

How the LSP Health Monitor works

Summary

The LSP Health Monitor feature provides the capability to proactively monitor Layer 3 MPLS VPNs. The general process for how the LSP Health Monitor works is as follows:

Workflow

1. The user configures an LSP Health Monitor operation and the BGP next hop neighbor discovery process is enabled.

   Configuring an LSP Health Monitor operation is similar to configuring a standard IP SLAs operation. To illustrate, all operation parameters for an LSP Health Monitor operation are configured after an identification number for the operation is specified. However, unlike standard IP SLAs operations, these configured parameters are then used as the base configuration for the individual IP SLAs LSP ping and LSP traceroute operations that will be created by the LSP Health Monitor. The LSP discovery process can potentially have a significant impact on the memory and CPU of the source PE device. To prevent unnecessary device performance issues, careful consideration should be taken when configuring the operational and scheduling parameters of an LSP Health Monitor operation. When the BGP next hop neighbor discovery process is enabled, a database of BGP next hop neighbors in use by any VRF associated with the source PE device is generated based on information from the local VRF and global routing tables. For more information about the BGP next hop neighbor discovery process, see the "Discovery of Neighboring PE Devices" section.

Note By default, only a single path between the source and destination PE devices is discovered. If the LSP discovery option is enabled, the equal-cost multipaths between the source and destination PE devices are discovered. For more information on how the LSP discovery process works, see the "LSP Discovery Process" section.

2. The user configures proactive threshold monitoring parameters for the LSP Health Monitor operation. For more information about proactive threshold monitoring, see the "Proactive Threshold Monitoring for the LSP Health Monitor" section. Depending on the proactive threshold monitoring configuration options chosen, SNMP trap notifications or syslog messages are generated as threshold violations are met.
3. The user configures multioperation scheduling parameters for the LSP Health Monitor operation. For more information about multioperation scheduling, see the "Multioperation Scheduling for the LSP Health Monitor" section.

   Once the LSP Health Monitor operation is started, a single IP SLAs operation is automatically created (based on parameters configured in Step 1) for each applicable PE (BGP next hop) neighbor. The IP SLAs

operations will measure network connectivity between the source PE device and the discovered destination PE device. The start time and frequency of each measurement is based on the multioperation scheduling parameters defined by the user.

Addition and Deletion of IP SLAs Operations

The LSP Health Monitor receives periodic notifications about BGP next hop neighbors that have been added to or removed from a particular VPN. This information is stored in a queue maintained by the LSP Health Monitor. Based on the information in the queue and user-specified time intervals, new IP SLAs operations are automatically created for newly discovered PE devices and existing IP SLAs operations are automatically deleted for any PE devices that are no longer valid. The automatic deletion of operations can be disabled. However, disabling this function is not recommended because these operations would then need to be deleted manually.

If the LSP discovery option is enabled, creation of LSP discovery groups for newly discovered BGP next hop neighbors will follow the same process as described in the "LSP Discovery Process" section. If a BGP next hop neighbor is removed from a particular VPN, all the corresponding LSP discovery groups and their associated individual IP SLAs operations and statistics are removed from the LSP discovery group database.

Access lists for filtering BGP next hop neighbors

Standard IP access lists can be configured to restrict the number of IP SLAs operations that are automatically created by the LSP Health Monitor. When the IP SLAs access list parameter is configured, the list of BGP next hop neighbors discovered by the LSP Health Monitor is filtered based on the conditions defined by the associated standard IP access list. In other words, the LSP Health Monitor will automatically create IP SLAs operations only for those BGP next hop neighbors with source addresses that satisfy the criteria permitted by the standard IP access list.

Unique identifier for each automatically created IP SLAs operation

The IP SLAs operations automatically created by the LSP Health Monitor are uniquely identified by their owner field. The owner field of an operation is generated using all the parameters that can be configured for that particular operation. If the length of the owner field is longer than 255 characters, it will be truncated.

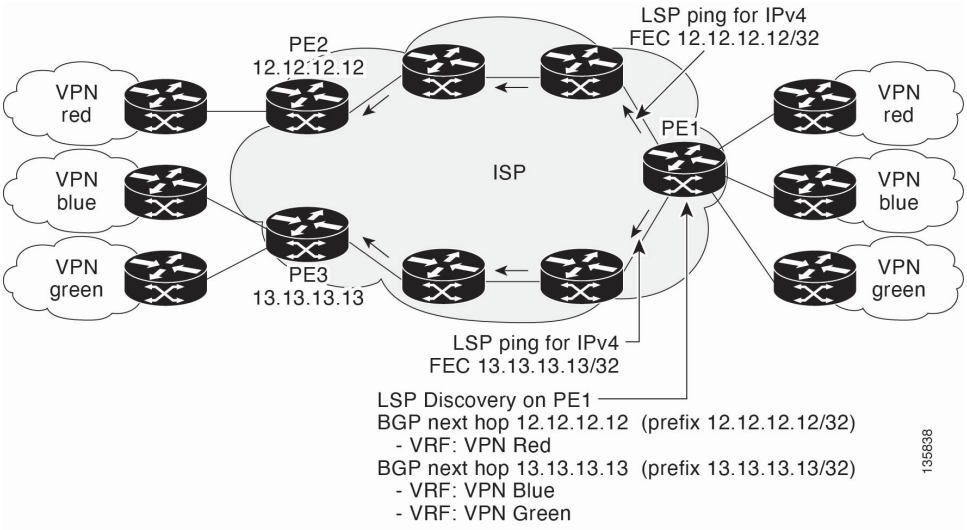
Discovery of neighboring PE devices

A BGP next hop neighbor discovery process is used to find the BGP next hop neighbors in use by any VRF associated with the source PE device. In most cases, these neighbors will be PE devices.

When the BGP next hop neighbor discovery process is enabled, a database of BGP next hop neighbors in use by any VRF associated with the source PE device is generated based on information from the local VRF and global routing tables. As routing updates are received, new BGP next hop neighbors are added to and deleted from the database immediately.

The figure below shows how the BGP next hop neighbor discovery process works for a simple VPN scenario for an Internet service provider (ISP). In this example, there are three VPNs associated with device PE1: red, blue, and green. From the perspective of device PE1, these VPNs are reachable remotely through BGP next hop neighbors PE2 (device ID: 12.12.12.12) and PE3 (device ID: 13.13.13.13). When the BGP next hop neighbor discovery process is enabled on device PE1, a database is generated based on the local VRF and global routing tables. The database in this example contains two BGP next hop device entries: PE2 12.12.12.12 and PE3 13.13.13.13. The routing entries are maintained per next hop device to distinguish which next hop devices belong within which particular VRF. For each next hop device entry, the IPv4 Forward Equivalence IP SLAs Configuration Guide 4 Configuring IP SLAs LSP Health Monitor Operations Discovery of Neighboring PE Devices Class (FEC) of the BGP next hop device in the global routing table is provided so that it can be used by the MPLS LSP ping operation.

Figure 8: BGP Next Hop Neighbor Discovery for a Simple VPN



LSP Discovery

The LSP discovery option of an LSP Health Monitor operation provides the capability to discover the equal-cost multipaths for carrying MPLS traffic between the source and destination PE devices. Network connectivity measurements can then be performed for each of the paths that were discovered.

The general process for LSP discovery is as follows:

1. BGP next hop neighbors are discovered using the BGP next hop neighbor discovery process. For more information about the BGP next hop neighbor discovery process, see the "Discovery of Neighboring PE Routers" section.

   Once the LSP Health Monitor operation is started, a single IP SLAs operation is automatically created for each applicable PE (BGP next hop) neighbor. Only a single path to each applicable PE neighbor is discovered during this initial step of the LSP discovery process. For each next hop neighbor, the LSP Health Monitor creates an LSP discovery group (that initially consists of only the one discovered path) and assigns the group with a unique identification number. For more information about LSP discovery groups, see the "LSP Discovery Groups" section.

2. An LSP discovery request is sent by the LSP Health Monitor to the LSP discovery subsystem for each applicable BGP next hop neighbor. For each next hop neighbor in which an appropriate response is

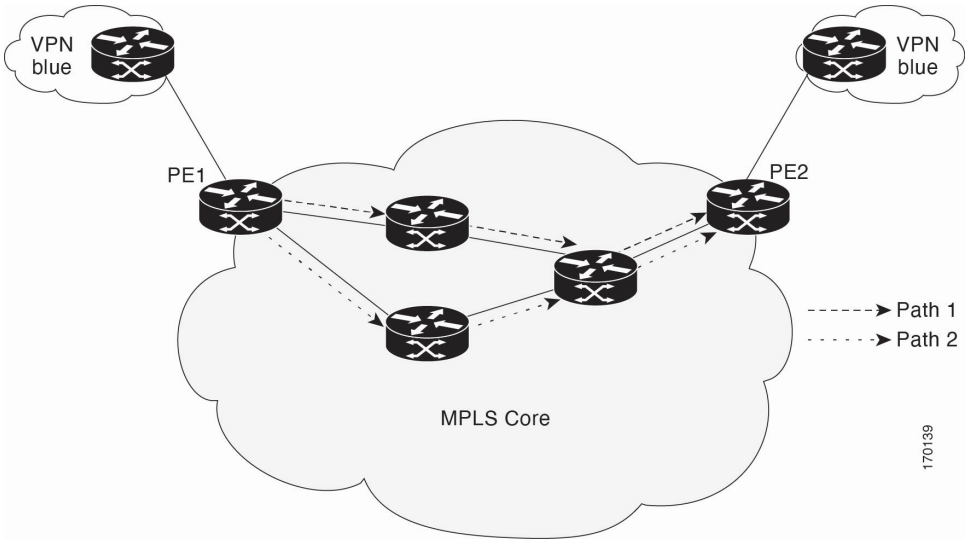
received, MPLS echo requests are sent one-by-one from the source PE device to discover the equal-cost multipaths. The parameters that uniquely identify each equal-cost multipath (127/8 destination IP address [LSP selector] and the PE outgoing interface) are added to the associated LSP discovery database.

Note For a given LSP Health Monitor operation, the user can define the maximum number of BGP next hop neighbors that can be concurrently undergoing LSP discovery.

3. Each individual IP SLAs operation (created for each applicable PE neighbor) uses an IP SLAs LSP ping superoperation to measure network connectivity across all equal-cost multipaths between the source PE device and discovered destination PE device. The IP SLAs superoperation operates by sending an LSP ping packet to the destination PE device and adjusting the LSP ping 127/8 LSP selector IP address for each discovered equal-cost multipath. For example, assume that there are three equal-cost multipaths to a destination PE device and the identified LSP selector IP addresses are 127.0.0.1, 127.0.0.5, and 127.0.0.6. The IP SLAs superoperation would sequentially send three LSP ping packets using the identified LSP selector IP addresses for directing the superoperation across the three paths. This technique ensures that there is only a single IP SLAs LSP ping operation for each source and destination PE device pair, and significantly reduces the number of active LSP ping operations sent by the source PE device.

The figure below illustrates a simple VPN scenario. This network consists of a core MPLS VPN with two PE devices (device PE1 and device PE2) belonging to the VRF named VPN blue. Suppose device PE1 is the source PE device for an LSP Health Monitor operation with the LSP discovery option enabled and that device PE2 is discovered by the BGP discovery process as a BGP next hop neighbor to device PE1. If path 1 and path 2 are equal-cost multipaths between device PE1 to device PE2, then the LSP discovery process would create an LSP discovery group consisting of path 1 and path 2. An IP SLAs LSP ping superoperation would also be created to monitor network availability across each path.

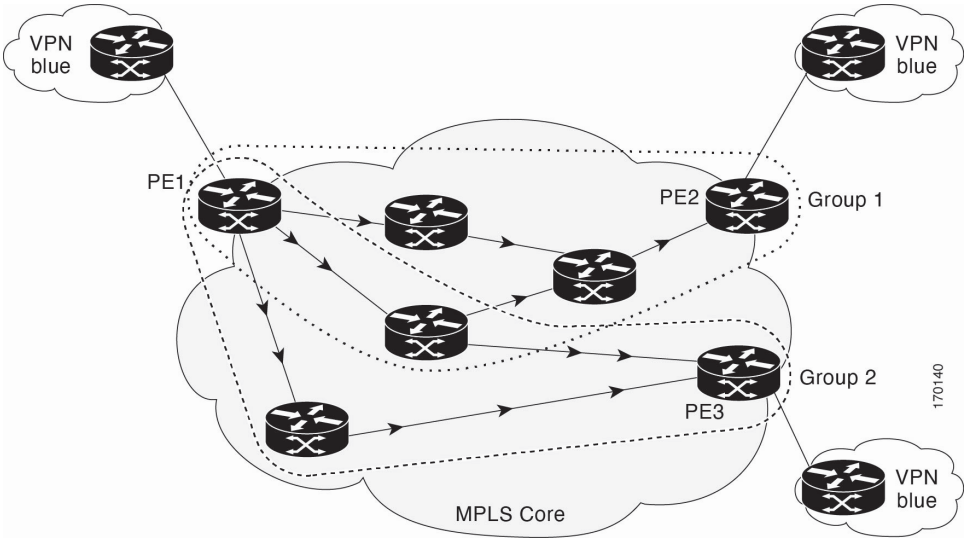
Figure 9: LSP Discovery for a Simple VPN



LSP Discovery Groups

A single LSP Health Monitor operation can be comprised of several LSP discovery groups depending on the number of BGP next hop neighbors discovered by the BGP next hop neighbor discovery process. Each LSP discovery group corresponds to one BGP next hop neighbor and is assigned a unique identification number (starting with the number 1). The figure below illustrates a simple VPN scenario. This network consists of a core MPLS VPN with three PE devices (device PE1, PE2, and PE3) belonging to the VRF named VPN blue. Suppose device PE1 is the source PE device for an LSP Health Monitor operation with the LSP discovery option enabled and that device PE2 and PE3 are discovered by the BGP discovery process as BGP next hop neighbors to device PE1. LSP discovery group 1 is created for the equal-cost multipaths between device PE1 to device PE2 and LSP discovery group 2 is created for the equal-cost multipaths between device PE1 to device PE3.

Figure 10: LSP Discovery Groups for a Simple VPN



Once the LSP Health Monitor operation is started, a single IP SLAs operation is automatically created for each applicable PE (BGP next hop) neighbor. Each IP SLAs operation (created for each applicable PE neighbor) uses an IP SLAs LSP ping superoperation to measure network connectivity across all equal-cost multipaths between the source PE device and discovered destination PE device. Each LSP ping superoperation corresponds to a single LSP discovery group.

The LSP ping superoperation operates by sending an LSP ping packet to the destination PE device and adjusting the LSP ping 127/8 LSP selector IP address for each discovered equal-cost multipath. The network connectivity statistics collected by each equal-cost multipath is aggregated and stored in one-hour increments (data can be collected for a maximum of two hours). Results are stored as group averages representative of all the equal-cost multipaths within the LSP discovery group for a given one-hour increment.

Each equal-cost multipath discovered between the source PE device and a BGP next hop neighbor is uniquely identified with the following parameters:

- 127/8 destination IP address (LSP selector) within the local host IP address range
- PE outgoing interface

The database for an LSP discovery group is updated if any of the following events occur:

- The corresponding LSP ping superoperation sends an LSP ping packet.
- An active equal-cost multipath is added to or deleted from the LSP discovery group.
- The user enters the Cisco command to delete all the aggregated statistical data for a particular LSP discovery group.

Proactive threshold monitoring for the LSP health monitor

Proactive threshold monitoring support for the LSP Health Monitor feature provides the capability for triggering SNMP trap notifications and syslog messages when user-defined reaction conditions (such as a connection loss or timeout) are met. Configuring threshold monitoring for an LSP Health Monitor operation is similar to configuring threshold monitoring for a standard IP SLAs operation.

LSP discovery option enabled

If the LSP discovery option for an LSP Health Monitor operation is enabled, SNMP trap notifications can be generated when one of the following events occurs:

- LSP discovery for a particular BGP next hop neighbor fails.
- Operational status of an LSP discovery group changes.

Possible reasons for which LSP discovery can fail for a particular BGP next hop neighbor are as follows:

- Expiration of time allowed for a BGP next hop neighbor to respond to an LSP discovery request.
- Return code is “Broken” or “Unexplorable” for all paths leading to the BGP next hop neighbor.

The table below describes the conditions for which the operational status of an LSP discovery group can change. Whenever an individual IP SLAs LSP ping operation of an LSP discovery group is executed, a return code is generated. Depending on the value of the return code and the current status of the LSP discovery group, the group status can change.

Table 1: Conditions for which an LSP discovery group status changes

Individual IP SLAs operation return code	Current group status = UP	Current group status = PARTIAL	Current group status = DOWN
OK	No group status change.	If return codes for all paths in the group are OK, then the group status changes to UP.	Group status changes to PARTIAL.
Broken or unexplorable	Group status changes to PARTIAL.	If return codes for all paths in the group are Broken or Unexplorable, then the group status changes to DOWN.	No group status change.

The return code for an individual IP SLAs LSP ping operation can be one of the following:

- OK: Indicates that the LSP is working properly. The customer VPN traffic will be sent across this path

- Broken: Indicates that the LSP is broken. Customer VPN traffic will not be sent across this path and may be discarded
- Unexplorable: Indicates that not all the paths to this PE neighbor have been discovered. This may be due to a disruption along the LSP or because the number of 127/8 IP addresses used for LSP selection has been exhausted.

The status of an LSP discovery group can be one of the following:

- UNKNOWN: Indicates that group status has not yet been determined and that the paths belonging to the group are in the process of being tested for the first time. Once this initial test is complete, the group status will change to UP, PARTIAL, or DOWN.
- UP: Indicates that all the paths within the group are active and no operation failures have been detected.
- PARTIAL: Indicates that an operation failure has been detected for one or more, but not all, of the paths within the group
- DOWN: Indicates that an operation failure has been detected for all the paths within the group

Secondary frequency option

With the introduction of the LSP Health Monitor feature, a new threshold monitoring parameter has been added that allows you to specify a secondary frequency. If the secondary frequency option is configured and a failure (such as a connection loss or timeout) is detected for a particular path, the frequency at which the path is remeasured will increase to the secondary frequency value (testing at a faster rate). When the configured reaction condition is met (such as N consecutive connection losses or N consecutive timeouts), an SNMP trap and syslog message can be sent and the measurement frequency will return to its original frequency value.

Multioperation scheduling for an LSP health monitor

Multioperation scheduling support for the LSP Health Monitor feature provides the capability to easily schedule the automatically created IP SLAs operations (for a given LSP Health Monitor operation) to begin at intervals equally distributed over a specified duration of time (schedule period) and to restart at a specified frequency. Multioperation scheduling is particularly useful in cases where the LSP Health Monitor is enabled on a source PE device that has a large number of PE neighbors and, therefore, a large number of IP SLAs operations running at the same time.

Newly created IP SLAs operations (for newly discovered BGP next hop neighbors) are added to the same schedule period as the operations that are currently running. To prevent too many operations from starting at the same time, the multioperation scheduling feature will schedule the operations to begin at random intervals uniformly distributed over the schedule period.

Configuring a multioperation schedule for an LSP Health Monitor is similar to configuring a standard multioperation schedule for a group of individual IP SLAs operations.

LSP discovery enabled

When a multioperation schedule for an LSP Health Monitor operation with LSP discovery is started, the BGP next hop neighbors are discovered, and network connectivity to each applicable neighbor is monitored using only a single LSP. Initially, network connectivity between the source PE device and discovered destination PE device is measured across only a single path. This initial condition is the same as if an LSP Health Monitor operation was performed without LSP discovery.

Specific information about the IP SLAs LSP ping operations that are created for newly discovered equal-cost paths during the succeeding iterations of the LSP discovery process are stored in the LSP discovery group database. These newly created IP SLAs LSP ping operations will start collecting data at the next iteration of network connectivity measurements for their associated LSP discovery group.

The start times for the individual IP SLAs LSP ping operations for each LSP discovery group is based on the number of LSP discovery groups and the schedule period of the multioperation schedule. For example, if three LSP discovery groups (Group 1, Group 2, and Group 3) are scheduled to run over a period of 60 seconds, the first LSP ping operation of Group 1 will start at 0 seconds, the first LSP ping operation of Group 2 will start at 20 seconds, and the first LSP ping operation of Group 3 will start at 40 seconds. The remaining individual IP SLAs LSP ping operations for each LSP discovery group will run sequentially after completion of the first LSP ping operation. For each LSP discovery group, only one LSP ping operation runs at a time.

Benefits of the LSP Health Monitor

- End-to-end LSP connectivity measurements across equal-cost multipaths for determining network availability or testing network connectivity in MPLS networks
- Proactive threshold monitoring through SNMP trap notifications and syslog messages
- Reduced network troubleshooting time for MPLS networks
- Scalable network error detection using fast retry capability
- Creation and deletion of IP SLAs operations based on network topology
- Discovery of Border Gateway Protocol (BGP) next hop neighbors based on local VPN routing and forwarding instances (VRFs) and global routing tables
- Multioperation scheduling of IP SLAs operations
- Pseudo-wire connectivity testing between MPLS network edges, with threshold violations and scalable operation scheduling
- Monitoring and SNMP trap alerts for round-trip time (RTT) threshold violations, connection loss, and command response timeouts

Guidelines to configure IP SLAs LSP Health Monitor operations

- The participating PE devices of an LSP Health Monitor operation must support the MPLS LSP ping feature. It is recommended that the Provider (P) devices also support the MPLS LSP Ping feature in order to obtain complete error reporting and diagnostics information.
- Ensure that the source PE device has enough memory to support the desired LSP Health Monitor functionality. Enabling the LSP discovery option can potentially have a significant impact on device memory. If there is not enough memory available during the LSP discovery process, the process will gracefully terminate and an error message will be displayed.

  The destination PE devices of an LSP Health Monitor operation do not require the IP SLAs Responder to be enabled.

- Once an LSP Health Monitor operation is started, its configuration parameters should not be changed until the operation has ended. Changing the configuration parameters while the operation is actively running could cause delays in obtaining network connectivity statistics.

Configure an LSP Health Monitor Operation

Follow the steps in each of these tasks to configure an LSP health monitor operation.

Procedure

Step 1 Perform any one of these tasks:

- Configure an LSP health monitor operation without LSP discovery on a PE device
- Configure the LSP health monitor operation with LSP discovery on a PE device

Step 2 Schedule IP SLAs operations

Step 3 Manually configure and schedule an IP SLAs LSP ping or LSP traceroute operation

Configure an LSP health monitor operation without LSP discovery on a PE device

Before you begin

If LSP discovery is disabled, only a single path between the source PE device and each BGP next hop neighbor is discovered.

Perform this task to configure an LSP health monitor operation without LSP discovery on a PE device.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **mpls discovery vpn next-hop**

Example:

```
Device(config)# mpls discovery vpn next-hop
```

(Optional) Enables the MPLS VPN BGP next hop neighbor discovery process.

Step 4 **mpls discovery vpn interval** *seconds*

Example:

```
Device(config)# mpls discovery vpn interval 120
```

(Optional) Specifies the time interval at which routing entries that are no longer valid are removed from the BGP next hop neighbor discovery database of an MPLS VPN.

Step 5 **auto ip sla mpls-lsp-monitor** *operation-number*

Example:

```
Device(config)# auto ip sla mpls-lsp-monitor 1
```

Begins configuration for an LSP Health Monitor operation and enters auto IP SLA MPLS configuration mode.

Entering this command automatically enables the **mpls discovery vpn next-hop** command.

Step 6 Choose one of the following:

- **type echo** [**ipsla-vrf-all** | **vrf** *vrf-name*]
- **type pathEcho** [**ipsla-vrf-all** | **vrf** *vrf-name*]

Example:

```
Device(config-auto-ip-sla-mpls)# type echo ipsla-vrf-all
OR
Device(config-auto-ip-sla-mpls)# type pathEcho ipsla-vrf-all
```

- **type echo** [**ipsla-vrf-all** | **vrf** *vrf-name*]:

  Enters MPLS parameters configuration submode and allows the user to configure the parameters for an IP SLAs LSP ping operation using the LSP Health Monitor.

- **type pathEcho** [**ipsla-vrf-all** | **vrf** *vrf-name*]:

  Enters MPLS parameters configuration submode and allows the user to configure the parameters for an IP SLAs LSP traceroute operation using the LSP Health Monitor.

Step 7 **access-list** *access-list-number*

Example:

```
Device(config-auto-ip-sla-mpls-params)# access-list 10
```

(Optional) Specifies the access list to apply to an LSP Health Monitor operation.

Step 8 **scan-interval** *minutes*

Example:

```
Device(config-auto-ip-sla-mpls-params)# scan-interval 5
```

(Optional) Sets the timer for the IP SLAs LSP Health Monitor database.

Step 9 **delete-scan-factor** *factor*

Example:

```
Device(config-auto-ip-sla-mpls-params)# delete-scan-factor 2
```

(Optional) Specifies the number of times the LSP Health Monitor should check the scan queue before automatically deleting IP SLAs operations for BGP next hop neighbors that are no longer valid.

- The default scan factor is 1. Each time the LSP Health Monitor checks the scan queue for updates, it deletes IP SLAs operations for BGP next hop neighbors that are no longer valid.
- If the scan factor is set to 0, IP SLAs operations will not be automatically deleted by the LSP Health Monitor. This configuration is not recommended.
- This command must be used with the scan-interval command.

Step 10 **force-explicit-null**

Example:

```
Device(config-auto-ip-sla-mpls-params)# force-explicit-null
```

(Optional) Adds an explicit null label to all echo request packets of an IP SLAs operation.

Step 11 **exp** *exp-bits*

Example:

```
Device(config-auto-ip-sla-mpls-params)# exp 5
```

(Optional) Specifies the experimental field value in the header for an echo request packet of an IP SLAs operation.

Step 12 **lsp-selector** *ip-address*

Example:

```
Device(config-auto-ip-sla-mpls-params)# lsp-selector 127.0.0.10
```

(Optional) Specifies the local host IP address used to select the LSP of an IP SLAs operation.

Step 13 **reply-dscp-bits** *dscp-value*

Example:

```
Device(config-auto-ip-sla-mpls-params)# reply-dscp-bits 5
```

(Optional) Specifies the differentiated services codepoint (DSCP) value for an echo reply packet of an IP SLAs operation.

Step 14 **reply-mode** {**ipv4** | **router-alert**}

Example:

```
Device(config-auto-ip-sla-mpls-params)# reply-mode router-alert
```

(Optional) Specifies the reply mode for an echo request packet of an IP SLAs operation.

The default reply mode is an IPv4 UDP packet.

Step 15 **request-data-size** *bytes*

Example:

```
Device(config-auto-ip-sla-mpls-params)# request-data-size 200
```

(Optional) Specifies the protocol data size for a request packet of an IP SLAs operation.

Step 16 **secondary-frequency** {**both** | **connection-loss** | **timeout**} *frequency*

Example:

```
Device(config-auto-ip-sla-mpls-params)# secondary-frequency connection-loss 10
```

(Optional) Sets the faster measurement frequency (secondary frequency) to which an IP SLAs operation should change when a reaction condition occurs.

Step 17 **tag** *text*

Example:

```
Device(config-auto-ip-sla-mpls-params)# tag testgroup
```

(Optional) Creates a user-specified identifier for an IP SLAs operation.

Step 18 **threshold** *milliseconds*

Example:

```
Device(config-auto-ip-sla-mpls-params)# threshold 6000
```

(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.

Step 19 **timeout** *milliseconds*

Example:

```
Device(config-auto-ip-sla-mpls-params)# timeout 7000
```

(Optional) Specifies the amount of time the IP SLAs operation waits for a response from its request packet.

Step 20 **ttl** *time-to-live*

Example:

```
Device(config-auto-ip-sla-mpls-params)# ttl 200
```

(Optional) Specifies the maximum hop count for an echo request packet of an IP SLAs operation.

Step 21 **exit**

Example:

```
Device(config-auto-ip-sla-mpls-params)# exit
```

Exits UDP configuration submode and returns to global configuration mode.

Step 22 **auto ip sla mpls-lsp-monitor reaction-configuration** *operation-number* **react** {**connectionLoss** | **timeout**} [**action-type** *option*] [**threshold-type** {**consecutive** [*occurrences*] | **immediate** | **never**}]

Example:

```
Device(config)# auto ip sla mpls-lsp-monitor reaction-configuration 1 react connectionLoss
 action-type trapOnly threshold-type consecutive 3
```

(Optional) Configures certain actions to occur based on events under the control of the LSP Health Monitor.

- *operation-number*: Specifies the LSP Health Monitor operation number to configure.
- **react** {**connectionLoss** | **timeout**}: Defines the monitored event type:
  - **connectionLoss**: Monitors one-way connection loss events.
  - **timeout:** Monitors one-way timeout events.

- **action-type** *option*: (Optional) Defines the action taken when a threshold violation occurs. Options include:
  - **none**: No action taken (default).
  - **trapOnly**: Sends an SNMP trap notification.
- **threshold-type**: (Optional) Defines when the action is triggered:
  - **consecutive** [*occurrences*]: Action occurs after a specified number of consecutive violations (default is 5, range 1-16).
  - **immediate**: Action occurs immediately upon violation.
  - **never**: No threshold violation monitoring (default).

Step 23 **exit**

Example:

```
Device(config)# exit
```

Exits global configuration mode and returns to privileged EXEC mode.

Configure the LSP health monitor operation with LSP discovery on a PE device

Before you begin

- The LSP Health Monitor with LSP Discovery feature supports Layer 3 MPLS VPNs only.
- The LSP discovery option does not support IP SLAs LSP traceroute operations.
- The LSP discovery option does not support IP SLAs VCCV operations.
- The LSP discovery process can potentially have a significant impact on the memory and CPU of the source PE device. To prevent unnecessary device performance issues, careful consideration should be taken when configuring the operational and scheduling parameters of an LSP Health Monitor operation

Perform this task to configure the LSP health monitor operation with LSP discovery on a PE device.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **mpls discovery vpn next-hop**

Example:

```
Device(config)# mpls discovery vpn next-hop
```

(Optional) Enables the MPLS VPN BGP next hop neighbor discovery process.

Step 4 **mpls discovery vpn interval** *seconds*

Example:

```
Device(config)# mpls discovery vpn interval 120
```

(Optional) Specifies the time interval at which routing entries that are no longer valid are removed from the BGP next hop neighbor discovery database of an MPLS VPN.

Step 5 **auto ip sla mpls-lsp-monitor** *operation-number*

Example:

```
Device(config)# auto ip sla mpls-lsp-monitor 1
```

Begins configuration for an LSP Health Monitor operation and enters auto IP SLA MPLS configuration mode.

Entering this command automatically enables the **mpls discovery vpn next-hop** command.

Step 6 **type echo** [**ipsla-vrf-all** | **vrf** *vrf-name*]

Example:

```
Device(config-auto-ip-sla-mpls)# type echo ipsla-vrf-all
```

Enters MPLS parameters configuration submode and allows the user to configure the parameters for an IP SLAs LSP ping operation using the LSP Health Monitor.

Step 7 Configure optional parameters for the IP SLAs LSP echo operation.

Step 8 **path-discover**

Example:

```
Device(config-auto-ip-sla-mpls-params)# path-discover
```

Enables the LSP discovery option for an IP SLAs LSP Health Monitor operation and enters LSP discovery parameters configuration submode.

Step 9 **hours-of-statistics-kept** *hours*

Example:

```
Device(config-auto-ip-sla-mpls-lpd-params)# hours-of-statistics-kept-1
```

(Optional) Sets the number of hours for which LSP discovery group statistics are maintained for an LSP Health Monitor operation.

Step 10 **force-explicit-null**

Example:

```
Device(config-auto-ip-sla-mpls-lpd-params)# force-explicit-null
```

(Optional) Adds an explicit null label to all echo request packets of an IP SLAs operation.

Step 11 **interval** *milliseconds*

Example:

```
Device(config-auto-ip-sla-mpls-lpd-params)# interval 2
```

(Optional) Specifies the time interval between MPLS echo requests that are sent as part of the LSP discovery process for an LSP Health Monitor operation.

Step 12 **lsp-selector-base** *ip-address*

Example:

```
Device(config-auto-ip-sla-mpls-lpd-params)# lsp-selector-base 127.0.0.10
```

(Optional) Specifies the base IP address used to select the LSPs belonging to the LSP discovery groups of an LSP Health Monitor operation.

Step 13 **maximum-sessions** *number*

Example:

```
Device(config-auto-ip-sla-mpls-lpd-params)# maximum-sessions 2
```

(Optional) Specifies the maximum number of BGP next hop neighbors that can be concurrently undergoing LSP discovery for a single LSP Health Monitor operation.

Note
Careful consideration should be used when configuring this parameter to avoid a negative impact on the device's CPU.

Step 14 **scan-period** *minutes*

Example:

```
Device(config-auto-ip-sla-mpls-lpd-params)# scan-period 30
```

(Optional) Sets the amount of time after which the LSP discovery process can restart for an LSP Health Monitor operation.

Step 15 **session-timeout** *seconds*

Example:

```
Device(config-auto-ip-sla-mpls-lpd-params)# session-timeout 60
```

(Optional) Sets the amount of time the LSP discovery process for an LSP Health Monitor operation waits for a response to its LSP discovery request for a particular BGP next hop neighbor.

Step 16 **timeout** *milliseconds*

Example:

```
Device(config-auto-ip-sla-mpls-lpd-params)# timeout 7000
```

(Optional) Specifies the amount of time the IP SLAs operation waits for a response from its request packet.

Step 17 **exit**

Example:

```
Device(config-auto-ip-sla-mpls-lpd-params)# exit
```

Exits LSP discovery parameters configuration submode and returns to MPLS parameters configuration mode.

Step 18 **exit**

Example:

```
Device(config-auto-ip-sla-mpls-params)# exit
```

Exits LSP discovery parameters configuration submode and returns to MPLS parameters configuration mode.

Step 19 **auto ip sla mpls-lsp-monitor reaction-configuration** *operation-number* **react lpd** {**lpd-group** [**retry** *number*] | **tree-trace**}[**action-type trapOnly**]

Example:

```
Device(config)# auto ip sla mpls-lsp-monitor reaction-configuration 1 react lpd lpd-group
retry 3 action-type trapOnly
```

(Optional) Configures the proactive threshold monitoring parameters for an LSP Health Monitor operation with LSP discovery enabled.

Step 20 **ip sla logging traps**

Example:

```
Device(config)# ip sla logging traps
```

(Optional) Enables the generation of SNMP system logging messages specific to IP SLAs trap notifications.

Step 21 **exit**

Example:

```
Device(config)# exit
```

Exits global configuration mode and returns to privileged EXEC mode.

Schedule LSP Health Monitor Operations

Before you begin

- All IP SLAs operations to be scheduled must be already configured.
- The LSP discovery process can potentially have a significant impact on the memory and CPU of the source PE device. Careful consideration should be taken when configuring the scheduling parameters to prevent too many IP SLAs LSP ping operations from running at the same time. The schedule period should be set to a relatively large value for large MPLS VPNs.
- Newly created IP SLAs operations (for newly discovered BGP next hop neighbors) are added to the same mulioperation schedule period as the operations that are currently running. To prevent too many operations from starting at the same time, the multioperation scheduler will schedule the operations to begin at random intervals uniformly distributed over the schedule period.

Perform this task to schedule lsp health monitor operations.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **auto ip sla mpls-lsp-monitor schedule** *operation-number* **schedule-period** *seconds* [**frequency** [*seconds*]] [**start-time** {**after** *hh:mm:ss* | *hh: mm*[*:ss*] [*month day* | *day month*] | **now** | **pending**}]

Example:

```
Device(config)# auto ip sla mpls-lsp-monitor schedule 1 schedule-period 60 start-time now
```

Configures the scheduling parameters for an LSP Health Monitor operation.

Step 4 **exit**

Example:

```
Device(config)# exit
```

Exits global configuration mode and returns to privileged EXEC mode.

Manually configure and schedule an IP SLAs LSP ping or LSP traceroute operation

Perform this task to manually configure and schedule an IP SLAs LSP ping or LSP traceroute operation.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla** *operation-number*

Example:

```
Device(config)# ip sla 10
```

Starts configuring an IP SLAs operation and enters IP SLA configuration mode.

Step 4 Choose one of the following:

- **mpls lsp ping ipv4** *destination-address destination-mask* [**force-explicit-null**] [**lsp-selector** *ip-address*] [**src-ip-addr** *source-address*] [**reply** {**dscp** *dscp-value* | **mode** {**ipv4** | **router-alert**}}]
- **mpls lsp trace ipv4** *destination-address destination-mask* [**force-explicit-null**] [**lsp-selector** *ip-address*] [**src-ip-addr** *source-address*] [**reply** {**dscp** *dscp-value* | **mode** {**ipv4** | **router-alert**}}]

Example:

```
Device(config-ip-sla)# mpls lsp ping ipv4 192.168.1.4 255.255.255.255 lsp-selector 127.1.1.1
OR
Device(config-ip-sla)# mpls lsp trace ipv4 192.168.1.4 255.255.255.255 lsp-selector 127.1.1.1
```

- **mpls lsp ping ipv4** *destination-address destination-mask* [**force-explicit-null**] [**lsp-selector** *ip-address*] [**src-ip-addr** *source-address*] [**reply** {**dscp** *dscp-value* | **mode** {**ipv4** | **router-alert**}}]:

  Configures the IP SLAs operation as an LSP ping operation and enters LSP ping configuration mode

- **mpls lsp trace ipv4** *destination-address destination-mask* [**force-explicit-null**] [**lsp-selector** *ip-address*] [**src-ip-addr** *source-address*] [**reply** {**dscp** *dscp-value* | **mode** {**ipv4** | **router-alert**}}]:

  Configures the IP SLAs operation as an LSP trace operation and enters LSP trace configuration mode.

Step 5 **exp** *exp-bits*

Example:

```
Device(config-sla-monitor-lspPing)# exp 5
```

(Optional) Specifies the experimental field value in the header for an echo request packet of an IP SLAs operation.

Note
The LSP ping configuration mode is used in this example and in the remaining steps. Except where noted, the same commands are also supported in the LSP trace configuration mode.

Step 6 **request-data-size** *bytes*

Example:

```
Device(config-auto-ip-sla-mpls-params)# request-data-size 200
```

(Optional) Specifies the protocol data size for a request packet of an IP SLAs operation.

Step 7 **secondary-frequency** {**connection-loss** | **timeout**} *frequency*

Example:

```
Device(config-sla-monitor-lspPing)# secondary-frequency connection-loss 10
```

(Optional) Sets the faster measurement frequency (secondary frequency) to which an IP SLAs operation should change when a reaction condition occurs.

Note
This command is for IP SLAs LSP ping operations only. LSP trace configuration mode does not support this command

Step 8 **tag** *text*

Example:

```
Device(onfig-sla-monitor-lspPing)# tag testgroup
```

(Optional) Creates a user-specified identifier for an IP SLAs operation.

Step 9 **threshold** *milliseconds*

Example:

```
Device(config-sla-monitor-lspPing)# threshold 6000
```

(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.

Step 10 **timeout** *milliseconds*

Example:

```
Device((config-sla-monitor-lspPing)# timeout 7000
```

(Optional) Specifies the amount of time the IP SLAs operation waits for a response from its request packet.

Step 11 **ttl** *time-to-live*

Example:

```
Device((config-sla-monitor-lspPing)# ttl 200
```

(Optional) Specifies the maximum hop count for an echo request packet of an IP SLAs operation.

Step 12 **exit**

Example:

```
Device(config-sla-monitor-lspPing)# exit
```

Exits UDP configuration submode and returns to global configuration mode.

Step 13 **ip sla reaction-configuration** *operation-number* [**react** *monitored-element*] [**threshold-type** {**never** | **immediate** | **consecutive** [*consecutive-occurrences*] | **xofy** [*x-value y-value*] | **average** [*number-of-probes*]}] [**threshold-value** *upper-threshold lower-threshold*] [**action-type** {**none** | **trapOnly** | **triggerOnly** | **trapAndTrigger**}]

Example:

```
Device(config)# ip sla reaction-configuration 1 react connectionLoss threshold-type
consecutive 3 action-type traponly
```

(Optional) Configures certain actions to occur based on events under the control of IP SLAs.

Step 14 **ip sla logging traps**

Example:

```
Device(config)# ip sla logging traps
```

(Optional) Enables the generation of SNMP system logging messages specific to IP SLAs trap notifications.

Step 15 **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh : mm*[: *ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh : mm : ss*}] [**ageout** *seconds*] [**recurring**]

Example:

```
Device(config)# ip sla schedule 1 start-time now
```

Configures the scheduling parameters for an IP SLAs operation.

Step 16 **exit**

Example:

```
Device(config)# exit
```

Exits global configuration submode and returns to privileged EXEC mode.

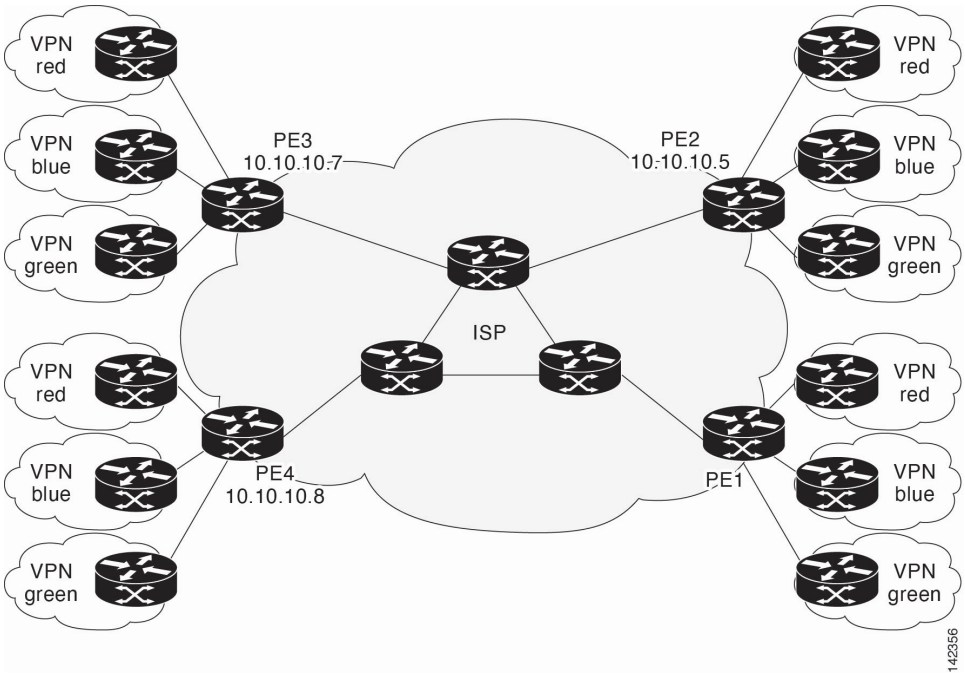
Configuration examples for LSP Health Monitors

These sections provide configuration examples for LSP health monitors.

Example: Configure and verify the LSP health monitor without LSP discovery

The figure below illustrates a simple VPN scenario for an ISP. This network consists of a core MPLS VPN with four PE devices belonging to three VPNs: red, blue, and green. From the perspective of device PE1, these VPNs are reachable remotely through BGP next hop devices PE2 (device ID: 10.10.10.5), PE3 (device ID: 10.10.10.7), and PE4 (device ID: 10.10.10.8).

Figure 11: Network Used for LSP Health Monitor Example



The following example shows how to configure operation parameters, proactive threshold monitoring, and scheduling options on PE1 (see the figure above) using the LSP Health Monitor. In this example, the LSP discovery option is enabled for LSP Health Monitor operation 1. Operation 1 is configured to automatically create IP SLAs LSP ping operations for all BGP next hop neighbors (PE2, PE3, and PE4) in use by all VRFs (red, blue, and green) associated with device PE1. The BGP next hop neighbor process is enabled, and the time interval at which routing entries that are no longer valid are removed from the BGP next hop neighbor

discovery database is set to 60 seconds. The time interval at which the LSP Health Monitor checks the scan queue for BGP next hop neighbor updates is set to 1 minute. The secondary frequency option is enabled for both connection loss and timeout events, and the secondary frequency is set to 10 seconds. As specified by the proactive threshold monitoring configuration, when three consecutive connection loss or timeout events occur, an SNMP trap notification is sent. Multioperation scheduling and the generation of IP SLAs SNMP system logging messages are enabled.

PE1 Configuration

```
Device> enable
Device# configure terminal
Device(config)# mpls discovery vpn interval 60
Device(config)# mpls discovery vpn next-hop
!
Device(config)# auto ip sla mpls-lsp-monitor 1
Device(config-auto-ip-sla-mpls)# type echo ipsla-vrf-all
Device(config-auto-ip-sla-mpls-params)# timeout 1000
Device(config-auto-ip-sla-mpls-params)# scan-interval 1
Device(config-auto-ip-sla-mpls-params)# secondary-frequency both 10
Device(config-auto-ip-sla-mpls-params)# exit
Device(config)# auto ip sla mpls-lsp-monitor reaction-configuration 1 react connectionLoss
 threshold-type
consecutive 3 action-type trapOnly
Device(config)# auto ip sla mpls-lsp-monitor reaction-configuration 1 react timeout
threshold-type consecutive
3 action-type trapOnly
Device(config)# ip sla traps
Device(config)# snmp-server enable traps rtr
!
Device(config)# auto ip sla mpls-lsp-monitor schedule 1 schedule-period 60 start-time now
```

The following is sample output from the **show ip sla mpls-lsp-monitor configuration** command for PE1:

```
PE1# show ip sla mpls-lsp-monitor configuration 1
Entry Number : 1
Modification time : *12:18:21.830 PDT Fri Aug 19 2005
Operation Type : echo
Vrf Name : ipsla-vrf-all
Tag :
EXP Value : 0
Timeout(ms) : 1000
Threshold(ms) : 5000
Frequency(sec) : Equals schedule period
LSP Selector : 127.0.0.1
ScanInterval(min) : 1
Delete Scan Factor : 1
Operations List : 100001-100003
Schedule Period(sec): 60
Request size : 100
Start Time : Start Time already passed
SNMP RowStatus : Active
TTL value : 255
Reply Mode : ipv4
Reply Dscp Bits :
Secondary Frequency : Enabled on Timeout
Value(sec) : 10
Reaction Configs :
Reaction : connectionLoss
Threshold Type : Consecutive
Threshold Count : 3
Action Type : Trap Only
Reaction : timeout
Threshold Type : Consecutive
```

```
Threshold Count : 3
Action Type : Trap Only
```

The following is sample output from the **show mpls discovery vpn** command for PE1:

```
PE1# show mpls discovery vpn
Refresh interval set to 60 seconds.
Next refresh in 46 seconds
Next hop 10.10.10.5 (Prefix: 10.10.10.5/32)
in use by: red, blue, green
Next hop 10.10.10.7 (Prefix: 10.10.10.7/32)
in use by: red, blue, green
Next hop 10.10.10.8 (Prefix: 10.10.10.8/32)
in use by: red, blue, green
```

The following is sample output from the **show ip sla mpls-lsp-monitor neighbors** command for PE1:

```
PE1# show ip sla mpls-lsp-monitor neighbors
IP SLA MPLS LSP Monitor Database : 1
BGP Next hop 10.10.10.5 (Prefix: 10.10.10.5/32) OK
ProbeID: 100001 (red, blue, green)
BGP Next hop 10.10.10.7 (Prefix: 10.10.10.7/32) OK
ProbeID: 100002 (red, blue, green)
BGP Next hop 10.10.10.8 (Prefix: 10.10.10.8/32) OK
ProbeID: 100003 (red, blue, green)
```

The following is sample output from the **show ip sla mpls-lsp-monitor scan-queue 1** and **debug ip sla mpls-lsp-monitor** commands when IP connectivity from PE1 to PE4 is lost. This output shows that connection loss to each of the VPNs associated with PE4 (red, blue, and green) was detected and that this information was added to the LSP Health Monitor scan queue. Also, since PE4 is no longer a valid BGP next hop neighbor, the IP SLAs operation for PE4 (Probe 10003) is being deleted.

```
PE1# show ip sla mpls-lsp-monitor scan-queue 1
Next scan Time after: 20 Secs
Next Delete scan Time after: 20 Secs
BGP Next hop Prefix vrf Add/Delete?
10.10.10.8 0.0.0.0/0 red Del(100003)
10.10.10.8 0.0.0.0/0 blue Del(100003)
10.10.10.8 0.0.0.0/0 green Del(100003)
```

```
PE1# debug ip sla mpls-lsp-monitor
IP SLAs MPLSLM debugging for all entries is on
*Aug 19 19:48: IP SLAs MPLSLM(1):Next hop 10.10.10.8 added in DeleteQ(1)
*Aug 19 19:49: IP SLAs MPLSLM(1):Removing vrf red from tree entry 10.10.10.8
*Aug 19 19:56: IP SLAs MPLSLM(1):Next hop 10.10.10.8 added in DeleteQ(1)
*Aug 19 19:56: IP SLAs MPLSLM(1):Next hop 10.10.10.8 added in DeleteQ(1)
*Aug 19 19:49: IP SLAs MPLSLM(1):Removing vrf blue from tree entry 10.10.10.8
*Aug 19 19:49: IP SLAs MPLSLM(1):Removing vrf green from tree entry 10.10.10.8
*Aug 19 19:49: IP SLAs MPLSLM(1):Removing Probe 100003
```

The following is sample output from the show ip sla mpls-lsp-monitor scan-queue 1 and debug ip sla mpls-lsp-monitor commands when IP connectivity from PE1 to PE4 is restored. This output shows that each of the VPNs associated with PE4 (red, blue, and green) were discovered and that this information was added to the LSP Health Monitor scan queue. Also, since PE4 is a newly discovered BGP next hop neighbor, a new IP SLAs operation for PE4 (Probe 100005) is being created and added to the LSP Health Monitor multioperation schedule. Even though PE4 belongs to three VPNs, only one IP SLAs operation is being created.

```
PE1# show ip sla mpls-lsp-monitor scan-queue 1
Next scan Time after: 23 Secs
Next Delete scan Time after: 23 Secs
BGP Next hop Prefix vrf Add/Delete?
10.10.10.8 10.10.10.8/32 red Add
```

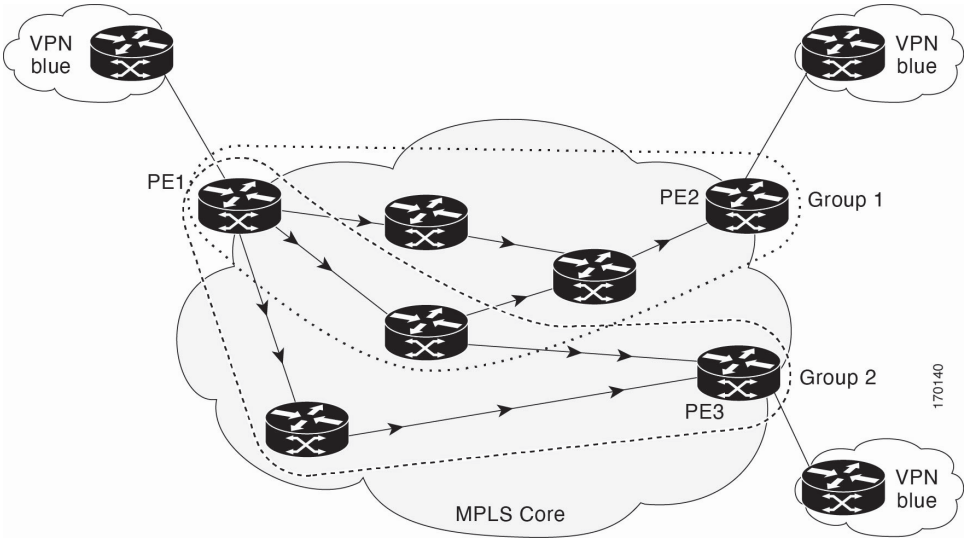
```
10.10.10.8 10.10.10.8/32 blue Add
10.10.10.8 10.10.10.8/32 green Add

PE1# debug ip sla mpls-lsp-monitor
IP SLAs MPLSLM debugging for all entries is on
*Aug 19 19:59: IP SLAs MPLSLM(1):Next hop 10.10.10.8 added in AddQ
*Aug 19 19:59: IP SLAs MPLSLM(1):Next hop 10.10.10.8 added in AddQ
*Aug 19 19:59: IP SLAs MPLSLM(1):Next hop 10.10.10.8 added in AddQ
*Aug 19 19:59: IP SLAs MPLSLM(1):Adding vrf red into tree entry 10.10.10.8
*Aug 19 19:59: IP SLAs MPLSLM(1):Adding Probe 100005
*Aug 19 19:59: IP SLAs MPLSLM(1):Adding ProbeID 100005 to tree entry 10.10.10.8 (1)
*Aug 19 19:59: IP SLAs MPLSLM(1):Adding vrf blue into tree entry 10.10.10.8
*Aug 19 19:59: IP SLAs MPLSLM(1):Duplicate in AddQ 10.10.10.8
*Aug 19 19:59: IP SLAs MPLSLM(1):Adding vrf green into tree entry 10.10.10.8
*Aug 19 19:59: IP SLAs MPLSLM(1):Duplicate in AddQ 10.10.10.8
*Aug 19 19:59: IP SLAs MPLSLM(1):Added Probe(s) 100005 will be scheduled after 26 secs over
schedule period 60
```

Example: Configure and verify the LSP health monitor with LSP discovery

The figure below illustrates a simple VPN scenario for an ISP. This network consists of a core MPLS VPN with two PE devices belonging to a VPN named red. From the perspective of device PE1, there are three equal-cost multipaths available to reach device PE2.

Figure 12: Network Used for LSP Health Monitor with LSP Discovery Example



The following example shows how to configure operation parameters, proactive threshold monitoring, and scheduling options on PE1 (see the figure above) using the LSP Health Monitor. In this example, the LSP discovery option is enabled for LSP Health Monitor operation 100. Operation 100 is configured to automatically create IP SLAs LSP ping operations for all equal-cost multipaths between PE1 and PE2. The BGP next hop neighbor process is enabled, and the time interval at which routing entries that are no longer valid are removed from the BGP next hop neighbor discovery database is set to 30 seconds. The time interval at which the LSP Health Monitor checks the scan queue for BGP next hop neighbor updates is set to 1 minute. The secondary frequency option is enabled for both connection loss and timeout events, and the secondary frequency is set to 5 seconds. The explicit null label option for echo request packets is enabled. The LSP rediscovery time period is set to 3 minutes. As specified by the proactive threshold monitoring configuration, an SNMP trap notification will be sent when an LSP discovery group status changes occurs. Multioperation scheduling and the generation of IP SLAs SNMP system logging messages are enabled.

PE1 Configuration

```
mpls discovery vpn next-hop
mpls discovery vpn interval 30
!
auto ip sla mpls-lsp-monitor 100
type echo ipsla-vrf-all
scan-interval 1
secondary-frequency both 5
!
path-discover
force-explicit-null
scan-period 3
!
auto ip sla mpls-lsp-monitor reaction-configuration 100 react lpd-group retry 3 action-type
trapOnly
!
auto ip sla mpls-lsp-monitor schedule 100 schedule-period 30 start-time now
!
ip sla logging traps
snmp-server enable traps rtr
```

The following is sample output from the **show ip sla mpls-lsp-monitor configuration** command for PE1:

```
PE1# show ip sla mpls-lsp-monitor configuration
Entry Number : 100
Modification time : *21:50:16.411 GMT Tue Jun 20 2006
Operation Type : echo
Vrf Name : ipsla-vrf-all
Tag :
EXP Value : 0
Timeout(ms) : 5000
Threshold(ms) : 50
Frequency(sec) : Equals schedule period
ScanInterval(min) : 1
Delete Scan Factor : 1
Operations List : 100002
Schedule Period(sec): 30
Request size : 100
Start Time : Start Time already passed
SNMP RowStatus : Active
TTL value : 255
Reply Mode : ipv4
Reply Dscp Bits :
Path Discover : Enable
Maximum sessions : 1
Session Timeout(seconds) : 120
Base LSP Selector : 127.0.0.0
Echo Timeout(seconds) : 5
Send Interval(msec) : 0
Label Shimming Mode : force-explicit-null
Number of Stats Hours : 2
Scan Period(minutes) : 3
Secondary Frequency : Enabled on Connection Loss and Timeout
Value(sec) : 5
Reaction Configs :
Reaction : Lpd Group
Retry Number : 3
Action Type : Trap Only
```

The following is sample output from the **show mpls discovery vpn** command for PE1:

```
PE1# show mpls discovery vpn
Refresh interval set to 30 seconds.
```

```
Next refresh in 4 seconds
Next hop 192.168.1.11
```

The following is sample output from the **show ip sla mpls-lsp-monitor neighbors** command for PE1:

```
PE1# show ip sla mpls-lsp-monitor neighbors
IP SLA MPLS LSP Monitor Database : 100
BGP Next hop 192.168.1.11 (Prefix: 192.168.1.11/32) OK Paths: 3
ProbeID: 100001 (red)
```

The following is sample output from the **show ip sla mpls-lsp-monitor lpd operational-state** command for LSP discovery group 100001:

```
PE1# show ip sla mpls-lsp-monitor lpd operational-state
Entry number: 100001
MPLSLM Entry Number: 100
Target FEC Type: LDP IPv4 prefix
Number of Statistic Hours Kept: 2
Last time LPD Stats were reset: *21:21:18.239 GMT Tue Jun 20 2006
Traps Type: 3
Latest Path Discovery Mode: rediscovery complete
Latest Path Discovery Start Time: *21:59:04.475 GMT Tue Jun 20 2006
Latest Path Discovery Return Code: OK
Latest Path Discovery Completion Time(ms): 3092
Number of Paths Discovered: 3
Path Information :
Path  Outgoing   Lsp        Link Conn Adj            Downstream
Index Interface  Selector   Type Id   Addr           Label Stack     Status
1      Et0/0     127.0.0.8  90   0    10.10.18.30    21               OK
2      Et0/0     127.0.0.2  90   0    10.10.18.30    21               OK
3      Et0/0     127.0.0.1  90   0    10.10.18.30    21               OK
```

The following is sample output from the **show ip sla mpls-lsp-monitor collection-statistics** command for LSP discovery group 100001:

```
PE1# show ip sla mpls-lsp-monitor collection-statistics
Entry number: 100001
Start Time Index: *21:52:59.795 GMT Tue Jun 20 2006
Path Discovery Start Time: *22:08:04.507 GMT Tue Jun 20 2006
Target Destination IP address: 192.168.1.11
Path Discovery Status: OK
Path Discovery Completion Time: 3052
Path Discovery Minimum Paths: 3
Path Discovery Maximum Paths: 3
LSP Group Index: 100002
LSP Group Status: up
Total Pass: 36
Total Timeout: 0 Total Fail: 0
Latest Probe Status: 'up,up,up'
Latest Path Identifier: '127.0.0.8-Et0/0-21,127.0.0.2-Et0/0-21,127.0.0.1-Et0/0-21'
Minimum RTT: 280 Maximum RTT: 324 Average RTT: 290
```

The following is sample output from the **show ip sla mpls-lsp-monitor summary** command for LSP Health Monitor operation 100:

```
PE1# show ip sla mpls-lsp-monitor summary 100
Index - MPLS LSP Monitor probe index
Destination - Target IP address of the BGP next hop
Status - LPD group status
LPD Group ID - Unique index to identify the LPD group
Last Operation Time - Last time an operation was attempted by
a particular probe in the LPD Group
Index Destination Status LPD Group ID Last Operation Time
100 192.168.1.11 up 100001 *22:20:29.471 GMT Tue Jun 20 2006
```

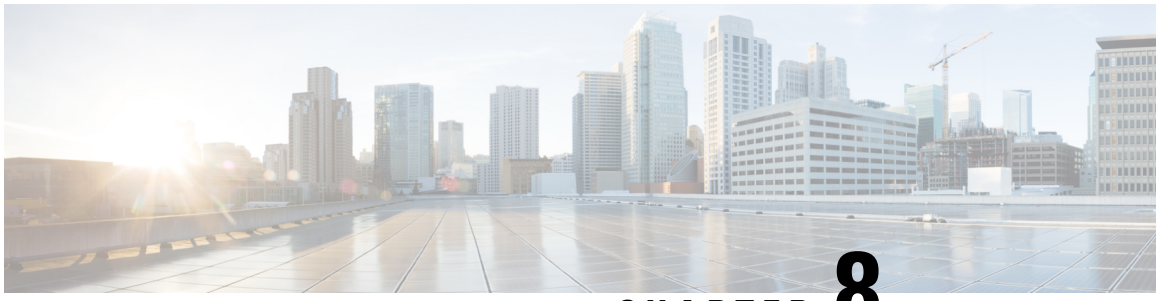
The following is sample output from the **show ip sla mpls-lsp-monitor summary** command for LSP discovery group 100001:

```
PE1# show ip sla mpls-lsp-monitor summary 100 group 100001
Group ID - unique number to identify a LPD group
Lsp-selector - Unique 127/8 address used to identify a LPD
Last Operation status - Latest probe status
Last RTT - Latest Round Trip Time
Last Operation Time - Time when the last operation was attempted
Group ID Lsp-Selector Status Failures Successes RTT Last Operation Time
100001 127.0.0.8 up 0 55 320 *22:20:29.471 GMT Tue
Jun 20 2006
100001 127.0.0.2 up 0 55 376 *22:20:29.851 GMT Tue
Jun 20 2006
100001 127.0.0.1 up 0 55 300 *22:20:30.531 GMT Tue
Jun 20 2006
```

Example: Manually configure an IP SLAs LSP ping operation

The following example shows how to manually configure and schedule an IP SLAs LSP ping operation:

```
Device> enable
Device# configure terminal
Device(config)# ip sla 1
Device(config)# mpls lsp ping ipv4 192.168.1.4 255.255.255.255 lsp-selector 127.1.1.1
Device(config)# frequency 120
Device(config)# secondary-frequency timeout 30
!
Device(config)# ip sla reaction-configuration 1 react connectionLoss threshold-type
consecutive 3 action-type
trapOnly
Device(config)# ip sla reaction-configuration 1 react timeout threshold-type consecutive 3
 action-type
trapOnly
Device(config)# ip sla logging traps
!
Device(config)# ip sla schedule 1 start-time now life forever
```



CHAPTER 8

IP SLAs Multi Operation Scheduler

Feature History for IP SLAs - Multi Operation Scheduler

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	IP SLAs - Multi Operation Scheduler: Multiple operations scheduling in IP SLAs allows you to schedule several IP SLAs operations simultaneously using a single command via the CLI or the CISCO-RTTMON-MIB.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

IP SLAs Multi Operation Scheduler

Multiple operations scheduling in IP SLAs allows you to schedule several IP SLAs operations simultaneously using a single command via the CLI or the CISCO-RTTMON-MIB.

Normal scheduling of IP SLAs operations allows scheduling only one operation at a time, which becomes inefficient and time-consuming when managing large networks with thousands of IP SLAs operations for monitoring network performance. This approach requires individually scheduling each operation, leading to significant manual effort and potential delays in comprehensive network monitoring.

IP SLAs Multioperation Scheduler allows scheduling multiple IP SLAs operations simultaneously as a group. This method reduces manual configuration, evenly distributes operations over a specified schedule period to minimize CPU load, and improves scalability and monitoring efficiency in large network environments

How IP SLAs Multioperation Scheduler works

Summary

IP SLAs Multioperation Scheduler requires specifying the operation ID numbers and the total time range over which these operations should start. It then automatically distributes the operations evenly at calculated intervals within that time frame. This even distribution helps control the amount of monitoring traffic, minimizes CPU utilization, and significantly enhances network scalability by preventing resource overload during operation start times

The IP SLAs multiple operations scheduling functionality allows to schedule multiple IP SLAs operations as a group, using the following configuration parameters:

- Group operation number: Group configuration or group schedule number of the IP SLAs operation to be scheduled.
- Operation ID numbers: A list of IP SLAs operation ID numbers in the scheduled operation group.
- Schedule period: The amount of time for which the IP SLAs operation group is scheduled.
- Ageout: The amount of time to keep the operation in memory when it is not actively collecting information.

  By default, the operation remains in memory indefinitely.
- Frequency: The amount of time after which each IP SLAs operation is restarted. When the frequency option is specified, it overwrites the operation frequency of all operations belonging to the group. Note that when the frequency option is not specified, the frequency for each operation is set to the value of theschedule period.
- Life: The amount of time the operation actively collects information. The operation can be configured to run indefinitely. By default, the lifetime of an operation is one hour.
- Start time: Time when the operation starts collecting information. You can specify an operation to start immediately or at an absolute start time using hours, minutes, seconds, day, and month.

The IP SLAs multiple operations scheduling functionality attempts to schedule the maximum number of operations possible without aborting. It automatically skips any IP SLAs operations that are already running or those that are not configured, that is it does not exist. Despite skipping these operations, the total number of operations considered is based on the number specified in the scheduling command, regardless of how many are missing or already running. When you schedule operations that are missing or already active, the

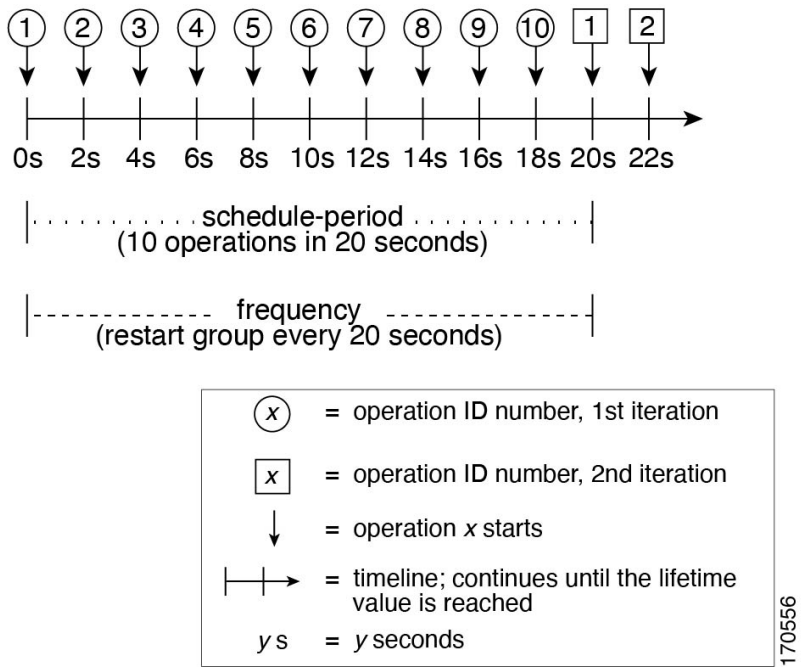
system displays a message indicating the number of active and missing operations. These messages appear only in such cases to inform you about the status of the operations being scheduled

For IP SLAs multiple operations scheduling, it is important that the operations you schedule are of the same type and have the same frequency. If you do not explicitly specify a frequency for the operations, the system uses the default frequency, which is set to be the same as the schedule period. The schedule period defines the total time frame during which all the specified operations should run. This ensures that the operations are evenly distributed and synchronized within the given time range, optimizing network performance monitoring and resource utilization

Default behavior of IP SLAs Multiple Operations Scheduler

The figure below illustrates the scheduling of operation group 1 that includes operation 1 to operation 10. Operation group 1 has a schedule period of 20 seconds, which means that all operations in the group will be started at equal intervals within a 20-second period. By default, the frequency is set to the same value as the configured schedule period. As shown in the figure below, configuring the frequencyis optional because 20 is the default.

Figure 13: Schedule period equals frequency: Default behavior



In this example, the first operation (operation 1) in operation group 1 will start at 0 seconds. All 10 operations in operation group 1 (operation 1 to operation 10) must be started in the schedule period of 20 seconds. The start time of each IP SLAs operation is evenly distributed over the schedule period by dividing the schedule period by the number of operations (20 seconds divided by 10 operations). Therefore, each operation will start 2 seconds after the previous operation.

The frequency in IP SLAs multiple operations scheduling defines the interval before the operation group starts again, that is how often the operations repeat. If you do not specify a frequency, it defaults to the value of the schedule period, which is the total time over which all specified operations are distributed and run. This means

the operations will repeat after the schedule period unless a different frequency is explicitly. In the example shown above, operation group 1 will start again every 20 seconds. This configuration provides optimal division (spacing) of operations over the specified schedule period.

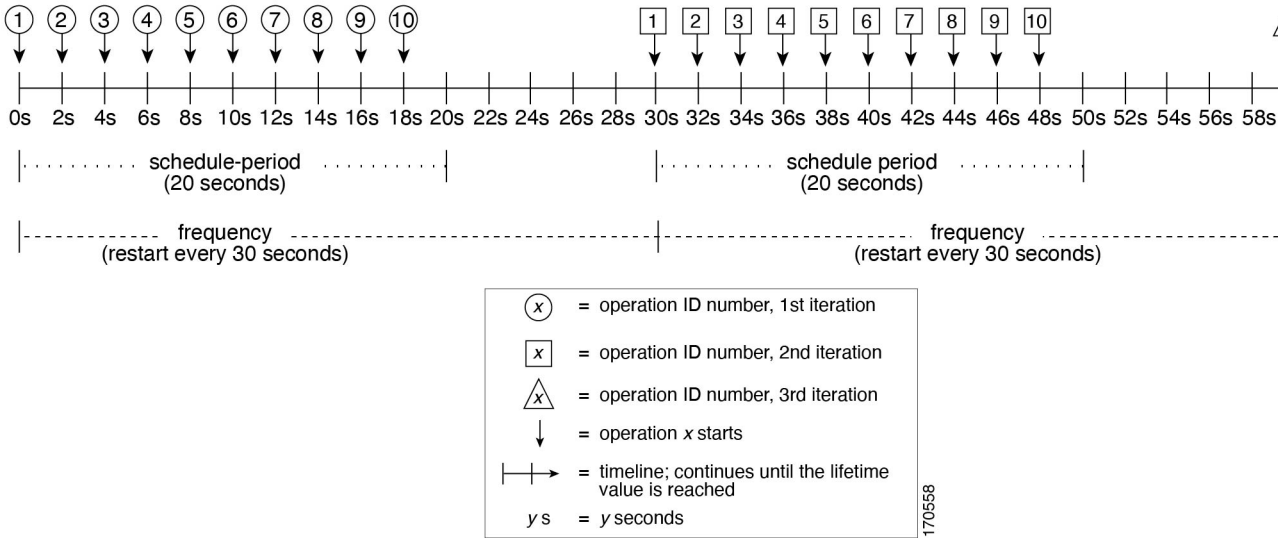
IP SLAs multiple Operations Scheduler with scheduling period less than frequency

The frequency value in IP SLAs multiple operations scheduling is the amount of time that passes before the entire schedule group is restarted. If the schedule period is less than the frequency, there will be a period of time during which no operations are started. This means that after all operations have run within the schedule period, there is a waiting interval before the next cycle begins, resulting in a temporary pause in operation starts.

The figure below illustrates the scheduling of operation 1 to operation 10 within operation group 2. Operation group 2 has a schedule period of 20 seconds and a frequency of 30 seconds.

Figure 14: Schedule period is less than frequency

ip sla group schedule 2 1-10 schedule-period 20 frequency 30



In this example, the first operation (operation 1) in operation group 2 will start at 0 seconds. All 10 operations in operation group 2 (operation 1 to operation 10) must be started in the schedule period of 20 seconds. The start time of each IP SLAs operation is evenly distributed over the schedule period by dividing the schedule period by the number of operations (20 seconds divided by 10 operations). Therefore, each operation will start 2 seconds after the previous operation.

In the first iteration of operation group 2, operation 1 starts at 0 seconds, and the last operation (operation 10) starts at 18 seconds. However, because the group frequency has been configured to 30 seconds each operation in the operation group is restarted every 30 seconds. So, after 18 seconds, there is a gap of 10 seconds as no operations are started in the time from 19 seconds to 29 seconds. Hence, at 30 seconds, the second iteration of operation group 2 starts. As all ten operations in the operation group 2 must start at an evenly distributed interval in the configured schedule period of 20 seconds, the last operation (operation 10) in the operation group 2 will always start 18 seconds after the first operation (operation 1).

As illustrated in the figure above, the following events occur:

- At 0 seconds, the first operation (operation 1) in operation group 2 is started.
- At 18 seconds, the last operation (operation 10) in operation group 2 is started. This means that the first iteration (schedule period) of operation group 1 ends here.
- From 19 to 29 seconds, no operations are started.
- At 30 seconds, the first operation (operation 1) in operation group 2 is started again. The second iteration of operation group 2 starts here.
- At 48 seconds (18 seconds after the second iteration started) the last operation (operation 10) in operation group 2 is started, and the second iteration of operation group 2 ends.
- At 60 seconds, the third iteration of operation group 2 starts.

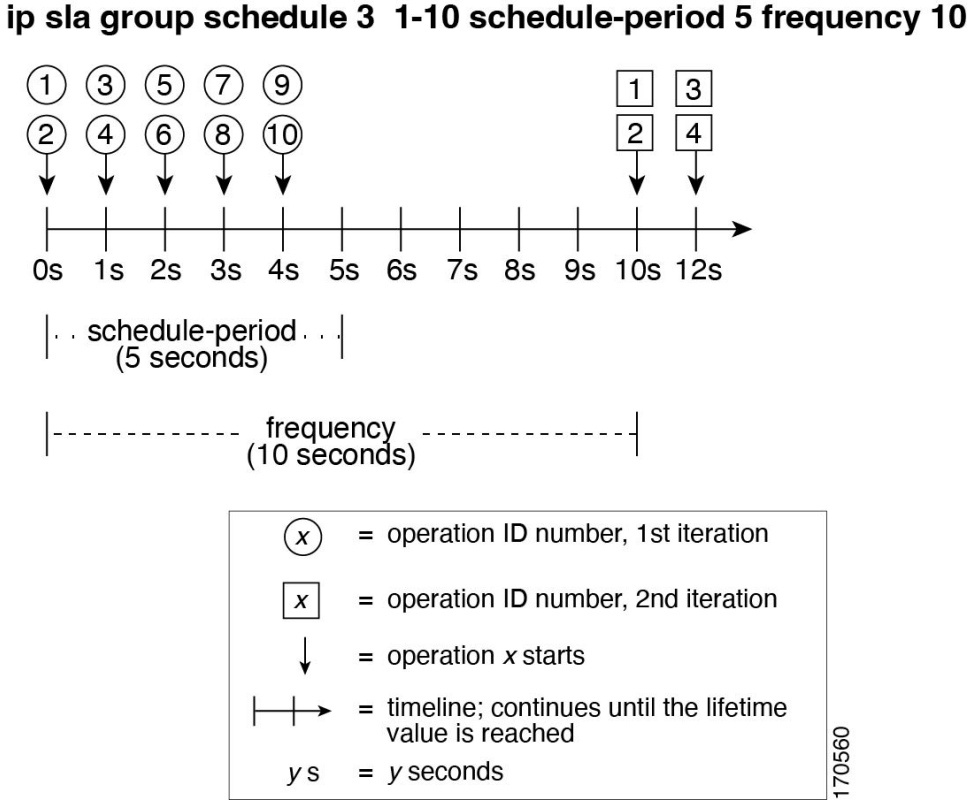
This process continues until the lifetime of operation group 2 ends. The lifetime value is configurable. The default lifetime for an operation group is forever.

Multiple operations scheduler: When the number of IP SLAs operations are greater than the schedule period

The minimum time interval between the start of IP SLAs operations in a multiple operations scheduling group is 1 second. When the number of operations to be scheduled exceeds the schedule period, the IP SLAs multiple operations scheduling functionality will schedule more than one operation to start within the same 1-second interval. If the number of operations does not divide evenly into 1-second intervals, the operations are distributed equally at the start of the schedule period, with any remaining operations scheduled to start at the last 1-second interval. This approach ensures that all operations are scheduled within the specified schedule period while respecting the minimum 1-second start interval constraint.

The figure below illustrates the scheduling of operation 1 to operation 10 within operation group 3. Operation group 3 has a schedule period of 5 seconds and a frequency of 10 seconds.

Figure 15: Number of IP SLAs operations Is greater than the schedule period-even distribution



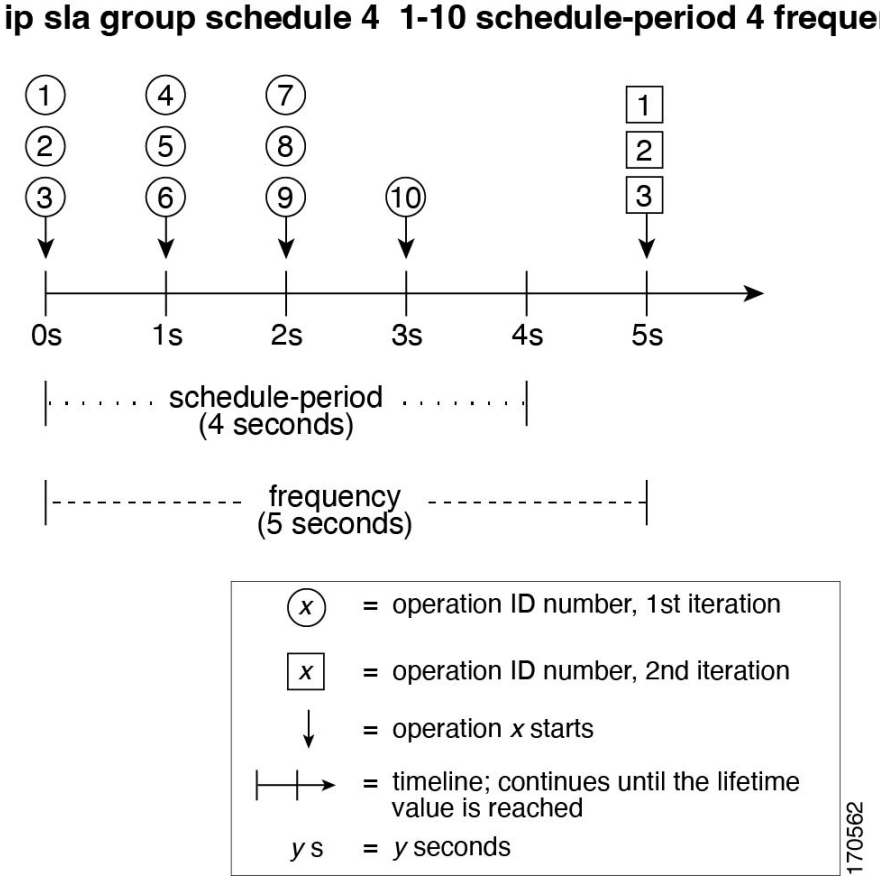
In this example, when dividing the schedule period by the number of operations (5 seconds divided by 10 operations, which equals one operation every 0.5 seconds) the start time of each IP SLAs operation is less than 1 second. Since the minimum time interval between the start of IP SLAs operations in a group operation is 1 second, the IP SLAs multiple operations scheduling functionality instead calculates how many operations it should start in each 1-second interval by dividing the number of operations by the schedule period (10 operations divided by 5 seconds). Therefore, as shown in the figure above, two operations will be started every 1 second.

As the frequency is set to 10 in this example, each iteration of operation group 3 will start 10 seconds after the start of the previous iteration. However, this distribution is not optimal as there is a gap of 5 seconds (frequency minus schedule period) between the cycles.

If the number of operations getting scheduled does not equally divide into 1-second intervals, then the operations are equally divided at the start of the schedule period with the remaining operations to start at the last 1-second interval.

The figure below illustrates the scheduling of operation 1 to operation 10 within operation group 4. Operation group 4 has a schedule period of 4 seconds and a frequency of 5 seconds.

Figure 16: Number of IP SLAs operations is greater than the schedule period: uneven distribution



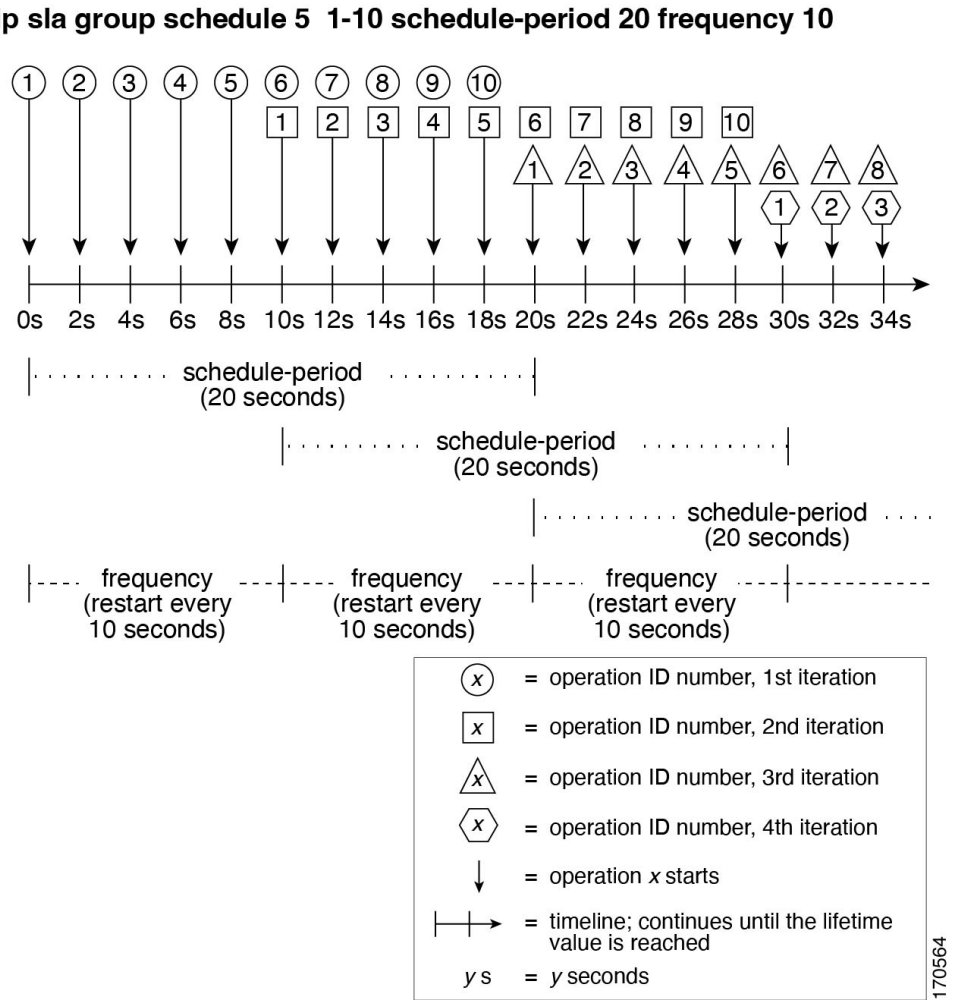
In this example, the IP SLAs multiple operations scheduling functionality calculates how many operations it should start in each 1-second interval by dividing the number of operations by the schedule period (10 operations divided by 4 seconds, which equals 2.5 operations every 1 second). Since the number of operations does not equally divide into 1-second intervals, this number will be rounded off to the next whole number (see the figure above) with the remaining operations to start at the last 1-second interval.

IP SLAs multiple operations scheduling with scheduling period greater than frequency

When the schedule period is greater than the frequency in IP SLAs multiple operations scheduling, the operations from one iteration of the operation group will overlap with the operations of the following iteration. This means that before the first group of operations has completed its schedule period, the next group starts, causing concurrent execution of operations from both iterations. This overlap can lead to increased load during the overlapping time frame, as multiple sets of operations run simultaneously. The frequency value determines how often the schedule group restarts, so if it is less than the schedule period, overlapping occurs accordingly.

The figure below illustrates the scheduling of operation 1 to operation 10 within operation group 5. Operation group 5 has a schedule period of 20 seconds and a frequency of 10 seconds.

Figure 17: IP SLAs group scheduling with schedule period greater than frequency



In this example, the first operation (operation 1) in operation group 5 will start at 0 seconds. All 10 operations in operation group 5 (operation 1 to operation 10) must be started in the schedule period of 20 seconds. The start time of each IP SLAs operation is evenly distributed over the schedule period by dividing the schedule period by the number of operations (20 seconds divided by 10 operations). Therefore, each operation will start 2 seconds after the previous operation.

In the first iteration of operation group 5, operation 1 starts at 0 seconds, and operation 10, the last operation in the operation group, starts at 18 seconds. Because the operation group is configured to restart every 10 seconds (frequency 10), the second iteration of operation group 5 starts again at 10 seconds, before the first iteration is completed. Therefore, an overlap of operations 6 to 10 of the first iteration occurs with operations 1 to 5 of the second iteration during the time period of 10 to 18 seconds (see the figure above). Similarly, there is an overlap of operations 6 to 10 of the second iteration with operations 1 to 5 of the third iteration during the time period of 20 to 28 seconds.

In this example, the start time of operation 1 and operation 6 need not be at exactly the same time, but will be within the same 2-second interval.

The configuration described in this section is not recommended as you can configure multiple operations to start within the same 1-second interval by configuring the number of operations greater than the schedule

period. For information, see the "Multiple Operations Scheduling When the Number of IP SLAs Operations Are Greater Than the Schedule Period" section.

IP SLAs random scheduler

The IP SLAs Random Scheduler feature is an enhancement to the existing IP SLAs multioperation scheduler.

The IP SLAs Multioperation Scheduling feature allows you to schedule multiple IP SLAs operations to start at equally distributed intervals over a specified time duration and to restart at a defined frequency. This helps in reducing network load by spreading out the operations evenly, improving monitoring consistency.

The IP SLAs Random Scheduler feature enhances this by scheduling multiple IP SLAs operations to start at random intervals uniformly distributed over the schedule period. Additionally, the operations restart at random frequencies uniformly distributed within a configured frequency range. This randomization improves the statistical accuracy of network performance assessments by avoiding synchronized operation starts that could mask transient network issues.

Note The IP SLAs random scheduler is not in compliance with RFC2330 because it does not account for inter-packet randomness.

The IP SLAs random scheduler option is disabled by default.

To enable the random scheduler option, you must set a frequency range when configuring a group schedule in global configuration mode. The group of operations restarts at uniformly distributed random frequencies within the specified frequency range. The following guidelines apply for setting the frequency range:

- The starting value of the frequency range should be greater than the timeout values of all the operations in the group operation.
- The starting value of the frequency range should be greater than the schedule period (amount of time for which the group operation is scheduled). This guideline ensures that the same operation does not get scheduled more than once within the schedule period.

The following guidelines apply if the random scheduler option is enabled:

- The individual operations in a group operation will be uniformly distributed to begin at random intervals over the schedule period.
- The group of operations restarts at uniformly distributed random frequencies within the specified frequency range.
- The minimum time interval between the start of each operation in a group operation is 100 milliseconds (0.1 seconds). If the random scheduler option is disabled, the minimum time interval is 1 second.
- Only one operation can be scheduled to begin at any given time. If the random scheduler option is disabled, multiple operations can begin at the same time.
- The first operation will always begin at 0 milliseconds of the schedule period.
- The order in which each operation in a group operation begins is random.

Benefit of IP SLAs Multiple Operations Scheduler

Scheduling multiple IP SLAs operations by distributing them equally over a scheduled period significantly reduces the load on the network and provides more consistent monitoring coverage. For example, if 60 operations are configured to start simultaneously within a 1-second interval over a 60-second schedule period, a network failure occurring 30 seconds after all operations have started might go undetected if the network is restored before the next start time. This is because all operations would have been active at the same time and then stopped, missing the failure window. However, if these 60 operations are evenly distributed at 1-second intervals over the 60-second period, some operations would be active during the failure, allowing detection. Conversely, if all operations run simultaneously and a failure occurs, all would fail, potentially exaggerating the severity of the failure. Thus, distributing operations helps achieve more accurate and reliable network performance monitoring by avoiding simultaneous operation starts and reducing false severity indications

Guidelines for IP SLAs Multioperation Scheduler

- Do not use the **no ip sla group schedule** and **ip sla group schedule** commands consecutively in a configuration file and copy it into the running configuration. This causes some of the Service Level Agreement (SLA) probes to go down.
- Configure the IP SLAs operations to be included in a group before scheduling the group.
- Determine the IP SLAs operations you want to schedule as a single group.
- Identify the network traffic type and the location of your network management station.
- Identify the topology and the types of devices in your network.
- Decide on the frequency of testing for each operation.

How to configure an IP SLAs multioperation scheduler

These sections provide configuration information on IP SLAs multioperation scheduler.

Schedule multiple IP SLAs operations

Before you begin

- All IP SLAs operations to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group should be the same.
- List of one or more operation ID numbers to be added to a multioperation group is limited to a maximum of 125 characters, including commas (,).

Perform this task to schedule multiple IP SLAs operations

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla group schedule** *group-operation-number operation-id-numbers* **schedule-period** *schedule-period-range* [**ageout** *seconds*] [**frequency** *group-operation-frequency*] [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm* [::*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*}]

Example:

```
Device(config)# ip sla group schedule 1 3,4,6-9 schedule-period 50 frequency range 80-100
```

Specifies an IP SLAs operation group number and the range of operation numbers to be scheduled in global configuration mode.

- *group-operation-number*: Identification number for the group of IP SLAs operations to be scheduled (range 0 to 65535).
- *operation-id-numbers*: List of one or more operation ID numbers to include in the multioperation schedule. You can specify individual IDs or ranges separated by commas (e.g., `3,4,6-9`). The total length can be up to 125 characters.
- **schedule-period** *schedule-period-range*: Specifies the amount of time (in seconds) over which the group of operations is scheduled to start. The schedule period defines the duration in which all specified operations should run.
- **ageout** *seconds*: Number of seconds to keep the IP SLAs operations in memory when they are not actively collecting information. Default is 0 (never ages out).
- **frequency** *group-operation-frequency*: Number of seconds after which each IP SLAs operation is restarted. This overrides the frequency of all operations in the group. The frequency can be a fixed number of seconds or a range for random scheduling, for example the range is from 80 to 100.
- **life** {**forever** | *seconds*} (optional): Specifies how long the operations will actively collect information.

  forever means indefinitely; otherwise, specify the number of seconds.
- **start-time** {*hh:mm*[:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*}: Specifies when the group of operations will start. Options include:
  - Absolute time with optional date, for example 13:01:30 Jul 15.
  - **pending**: This is the default. No data collection

- **now**: Starts immediately.
- **after** *hh:mm:ss*: Starts after a delay from the command entry time)

Step 4 **exit**

Example:

```
Device(config)# exit
```

Returns to the privileged EXEC mode.

Step 5 **show ip sla group schedule**

Example:

```
Device# show ip sla group schedule
```

(Optional) Displays the IP SLAs group schedule details.

Step 6 **show ip sla configuration**

Example:

```
Device# show ip sla configuration
```

(Optional) Displays the IP SLAs configuration details.

Enable the IP SLAs random scheduler

Peform this task to enable the IP SLAs randon scheduler.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla group schedule** *group-operation-number operation-id-numbers* **schedule-period** *schedule-period-range* [**ageout** *seconds*] [**frequency** *group-operation-frequency*] [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm* [*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*}]

Example:

```
Device(config)# ip sla group schedule 1 3,4,6-9 schedule-period 50 frequency range 80-100
```

Specifies an IP SLAs operation group number and the range of operation numbers to be scheduled in global configuration mode.

- *group-operation-number*: Identification number for the group of IP SLAs operations to be scheduled (range 0 to 65535).
- *operation-id-numbers*: List of one or more operation ID numbers to include in the multioperation schedule. You can specify individual IDs or ranges separated by commas (e.g., `3,4,6-9`). The total length can be up to 125 characters.
- **schedule-period** *schedule-period-range*: Specifies the amount of time (in seconds) over which the group of operations is scheduled to start. The schedule period defines the duration in which all specified operations should run.
- **ageout** *seconds*: Number of seconds to keep the IP SLAs operations in memory when they are not actively collecting information. Default is 0 (never ages out).
- **frequency** *group-operation-frequency*: Number of seconds after which each IP SLAs operation is restarted. This overrides the frequency of all operations in the group. The frequency can be a fixed number of seconds or a range for random scheduling, for example the range is from 80 to 100.
- **life** {**forever** | *seconds*} (optional): Specifies how long the operations will actively collect information.

  forever means indefinitely; otherwise, specify the number of seconds.
- **start-time** {*hh:mm*[:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*}: Specifies when the group of operations will start. Options include:
  - Absolute time with optional date, for example 13:01:30 Jul 15.
  - **pending**: This is the default. No data collection
  - **now**: Starts immediately.
  - **after** *hh:mm:ss*: Starts after a delay from the command entry time)

Step 4 **exit**

Example:

```
Device(config)# exit
```

Returns to the privileged EXEC mode.

Configuration examples for an IP SLAs multi operation scheduler

The following sections provide configuration examples for an IP SLAs multioperation scheduler.

Example: Enable the IP SLAs random scheduler

The following example shows how to schedule IP SLAs operations 1 to 3 as a group (identified as group 2). In this example, the operations are scheduled to begin at uniformly distributed random intervals over a schedule period of 50 seconds. The first operation is scheduled to start immediately. The interval is chosen from the specified range upon every invocation of the probe. The random scheduler option is enabled and the uniformly

distributed random frequencies at which the group of operations will restart is chosen within the range of 80-100 seconds.

```
Device> enable
Device# configure terminal
Device(config)# ip sla group schedule 2 1-3 schedule-period 50 frequency range 80-100
start-time now
```

Example: Schedule multiple IP SLAs operations

The following example shows how to scheduls IP SLAs operations 1 to 10 in the operation group 1 with a schedule period of 20 seconds. By default, the frequency is equivalent to the schedule period.

```
Device> enable
Device# configure terminal
Device(config)# ip sla group schedule 1 1-10 schedule-period 20
```

Example: Verify the IP SLAs multioperation schedulers

The following example shows the details of the scheduled multiple IP SLAs operation using the **show ip sla group schedule** command. The last line in the example indicates that the IP SLAs operations are multiple scheduled (TRUE).

```
Device# show ip sla group schedule
Multi-Scheduling Configuration:
Group Entry Number: 1
Probes to be scheduled: 1-10
Schedule period :20
Group operation frequency: 20
Multi-scheduled: TRUE
```

Example: Schedule multiple IP SLAs operations

The following example schedules IP SLAs operations 1 through 20 in the operation group 1 with a schedule period of 60 seconds and a life value of 1200 seconds. By default, the frequency is equivalent to the schedule period. In this example, the start interval is 3 seconds (schedule period divided by number of operations).

```
Device> enable
Device# configure terminal
Device(config)# ip sla group schedule 1 1-20 schedule-period 60 life 1200
```

Example: Verify the IP SLAs multioperation schedulers

The following example shows the details of the scheduled multiple IP SLAs operation using the **show ip sla group schedule** command.

```
Device# show ip sla group schedule
Group Entry Number: 1
Probes to be scheduled: 1-20
Total number of probes: 20
Schedule period: 60
Group operation frequency: Equals schedule period
Status of entry (SNMP RowStatus): Active
Next Scheduled Start Time: Start Time already passed
Life (seconds): 1200
Entry Ageout (seconds): never
```

The following example shows the details of the scheduled multiple IP SLAs operation using the **show ip sla configuration** command. The last line in the example indicates that the IP SLAs operations are multiple scheduled (TRUE).

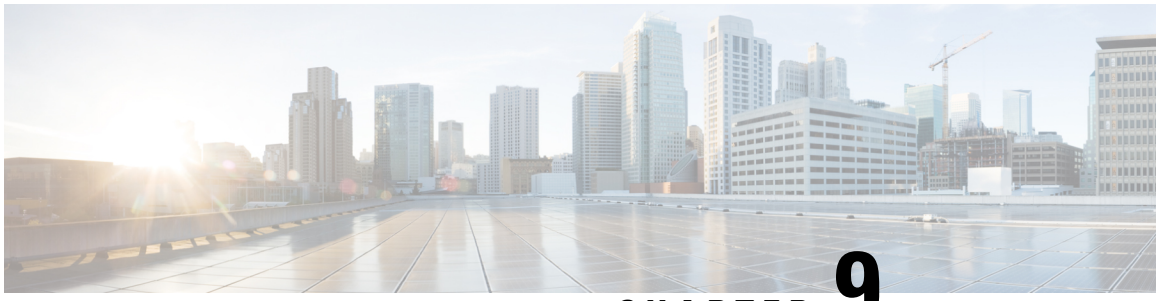
```
Device# show ip sla configuration 1
Entry number: 1
Owner:
Tag:
Type of operation to perform: udpEcho
Target address: 10.2.31.121
Source address: 0.0.0.0
Target port: 9001
Source port: 0
Request size (ARR data portion): 16
Operation timeout (milliseconds): 5000
Type Of Service parameters: 0x0
Verify data: No
Data pattern:
Vrf Name:
Control Packets: enabled
Operation frequency (seconds): 60
Next Scheduled Start Time: Start Time already passed
Life (seconds): 1200
Entry Ageout (seconds): never
Recurring (Starting Everyday): FALSE
Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000
Number of statistic hours kept: 2
Number of statistic distribution buckets kept: 1
Statistic distribution interval (milliseconds): 20
Enhanced History:
Number of history Lives kept: 0
Number of history Buckets kept: 15
History Filter Type: N
Group Scheduled : TRUE
```

The following example shows the latest operation start time of the scheduled multiple IP SLAs operation, when the operations are scheduled at equal intervals, using the show ip sla statistics command:

```
Device# show ip sla statistics | include Latest operation start time
Latest operation start time: *03:06:21.760 UTC Tue Oct 21 2003
Latest operation start time: *03:06:24.754 UTC Tue Oct 21 2003
Latest operation start time: *03:06:27.751 UTC Tue Oct 21 2003
Latest operation start time: *03:06:30.752 UTC Tue Oct 21 2003
Latest operation start time: *03:06:33.754 UTC Tue Oct 21 2003
Latest operation start time: *03:06:36.755 UTC Tue Oct 21 2003
Latest operation start time: *03:06:39.752 UTC Tue Oct 21 2003
Latest operation start time: *03:06:42.753 UTC Tue Oct 21 2003
Latest operation start time: *03:06:45.755 UTC Tue Oct 21 2003
Latest operation start time: *03:06:48.752 UTC Tue Oct 21 2003
Latest operation start time: *03:06:51.753 UTC Tue Oct 21 2003
Latest operation start time: *03:06:54.755 UTC Tue Oct 21 2003
Latest operation start time: *03:06:57.752 UTC Tue Oct 21 2003
Latest operation start time: *03:07:00.753 UTC Tue Oct 21 2003
Latest operation start time: *03:07:03.754 UTC Tue Oct 21 2003
Latest operation start time: *03:07:06.752 UTC Tue Oct 21 2003
Latest operation start time: *03:07:09.752 UTC Tue Oct 21 2003
Latest operation start time: *03:07:12.753 UTC Tue Oct 21 2003
Latest operation start time: *03:07:15.755 UTC Tue Oct 21 2003
Latest operation start time: *03:07:18.752 UTC Tue Oct 21 2003
```

Verify IP SLAs multiple operation scheduler

Command	Purpose
show ip sla statistics	Displays the IP SLAs operation details.
show ip sla group schedule	Displays the IP SLAs group schedule details.
show ip sla configuration	Displays the IP SLAs configuration details.



CHAPTER 9

IP SLAs TCP Connect Operation

Feature History for IP SLAs - TCP Connect Operation

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	IP SLAs - TCP Connect Operation: This operation measures the response time needed to establish a TCP connection between a Cisco device and a target device across an IP network.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

IP SLA TCP connect

The IP SLAs TCP Connect operation is used to measure the response time needed to establish a TCP connection between a Cisco device and a target device across an IP network. By utilizing TCP, a Layer 4 protocol known for providing reliable and full-duplex communication, this operation offers valuable insights into the performance and availability of network services. The target device can be any IP-enabled system, such as a server or a network appliance, or it can be configured as an IP SLAs Responder to enable additional features and more precise measurements. This capability enables network administrators to effectively monitor, analyze, and troubleshoot network performance by testing connectivity to various services and applications throughout the network.

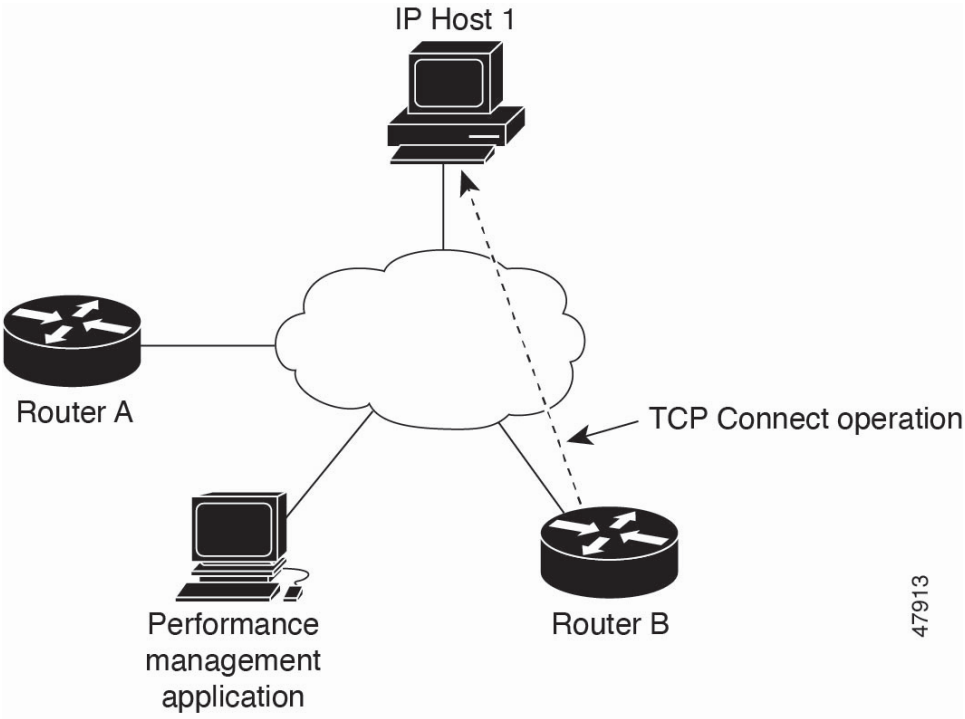
How IP SLA TCP works

Summary

In the figure below Device B is configured as the source IP SLAs device and a TCP Connect operation is configured with the destination device as IP Host 1.

Workflow

Figure 18: TCP Connect Operation



Connection response time is computed by measuring the time taken between sending a TCP request message from Device B to IP Host 1 and receiving a reply from IP Host 1.

IP SLAs TCP connect and IP SLAs responder

The accuracy of TCP Connect measurements is enhanced when the IP SLAs Responder is enabled on the destination Cisco device. When targeting another Cisco device, IP SLAs can initiate a TCP connection to any specified port, allowing for flexible and precise testing. However, if the destination is not a Cisco IP host, you need to specify a well-known port number corresponding to the intended service, such as port 21 for FTP, port 23 for Telnet, or port 80 for an HTTP server. This ensures that the operation can successfully establish a TCP connection and accurately measure network response times.

The use of the IP SLAs Responder is optional for TCP Connect operations when the target is a Cisco device, but it is not supported on non-Cisco devices. TCP Connect is typically used to test the availability of virtual circuits or applications by simulating connections to services like Telnet, SQL, and others. This approach allows network administrators to assess the performance and responsiveness of servers and applications, helping to verify and maintain agreed-upon IP service levels within the network.

Configure and scheduling a TCP connect operation on the source device

You can either configure a basic TCP connect operation or a TCP connect operation with optional parameters on a source device.

Perform only one of the following tasks:

Configure a basic TCP connect operation on the source device

Perform this procedure to configure a basic TCP connect operation on the source device.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla** *operation-number*

Example:

```
Device(config)# ip sla 10
```

Starts configuring an IP SLAs operation and enters IP SLA configuration mode.

Step 4 **tcp-connect** {*destination-ip-addres* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}]

Example:

```
Device(config-ip-sla)# tcp-connect 172.29.139.134 5000
```

Defines a TCP Connect operation and enters IP SLA TCP configuration mode.

Use the **control disable** keyword combination only if you disable the IP SLAs control protocol on both the source and target devices

- *destination-ip-address* | *destination-hostname*: Specifies the IP address or hostname of the target device for the UDP Jitter operation.
- *destination-port*: The UDP port number on the target device that will receive the packets.

- **source-ip** {*ip-address* | *hostname*}: (Optional) Specifies the source IP address or hostname from which packets will be sent.
- **source-port** *port-number*: (Optional) Specifies the UDP source port number for the test packets.
- **control** {**enable** | **disable**}: (Optional) Enables or disables the control protocol, which is used to notify the responder on the target device about the test.

  Use the **control disable** keyword combination only if you disable the IP SLAs control protocol on both source and destination devices.

Step 5 **frequency** *seconds*

Example:

```
Device(config-ip-sla)# frequency 30
```

(Optional) Sets the rate at which a specified IP SLAs operation repeats.

Step 6 **end**

Example:

```
Device(config-ip-sla)# end
```

Returns to privileged EXEC mode.

Configure a TCP connect operation with optional parameters on the source device

Perform this task to configure TCP connect operation with optional parameters on the source device.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla** *operation-number*

Example:

```
Device(config)# ip sla 10
```

Starts configuring an IP SLAs operation and enters IP SLA configuration mode.

Step 4 **tcp-connect** {*destination-ip-addres* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}]

Example:

```
Device(config-ip-sla)# tcp-connect 172.29.139.134 5000
```

Defines a TCP Connect operation and enters IP SLA TCP configuration mode.

Use the **control disable** keyword combination only if you disable the IP SLAs control protocol on both the source and target devices

- *destination-ip-address* | *destination-hostname*: Specifies the IP address or hostname of the target device for the UDP Jitter operation.
- *destination-port*: The UDP port number on the target device that will receive the packets.
- **source-ip** {*ip-address* | *hostname*}: (Optional) Specifies the source IP address or hostname from which packets will be sent.
- **source-port** *port-number*: (Optional) Specifies the UDP source port number for the test packets.
- **control** {**enable** | **disable**}: (Optional) Enables or disables the control protocol, which is used to notify the responder on the target device about the test.

  Use the **control disable** keyword combination only if you disable the IP SLAs control protocol on both source and destination devices.

Step 5 **history buckets-kept** *size*

Example:

```
Device(config-ip-sla-tcp)# history buckets-kept 25
```

(Optional) Sets the number of history buckets that are kept during the lifetime of an IP SLAs operation.

Step 6 **history distributions-of-statistics-kept** *size*

Example:

```
Device(config-ip-sla-tcp)# history distributions-of-statistics-kept 5
```

(Optional) Sets the number of statistics distributions kept per hop for an IP SLAs operation.

size: The range is from 1 to 20.

Step 7 **history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*]

Example:

```
Device(config-ip-sla-tcp)# history enhanced interval 900 buckets 100
```

(Optional) Enables enhanced history gathering for an IP SLAs operation.

- **interval** *seconds*: (Optional) The interval, in seconds, at which to collect and store enhanced statistics. The range is from 1 to 3600 seconds.
- **buckets** *number-of-buckets*: (Optional) The number of enhanced history buckets to retain. The range is from 1 to 100.

Step 8 **history filter** {**none** | **all** | **overThreshold** | **failures**}

Example:

```
Device(config-ip-sla-tcp)# history filter failures
```

(Optional) Defines the type of information kept in the history table for an IP SLAs operation.

Step 9 **frequency** *seconds*

Example:

```
Device(config-ip-sla-tcp)# frequency 30
```

(Optional) Sets the rate at which a specified IP SLAs operation repeats.

Step 10 **history hours-of-statistics-kept** *hours*

Example:

```
Device(config-ip-sla-tcp)# history hours-of-statistics-kept 4
```

(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.

Step 11 **history lives-kept** *lives*

Example:

```
Device(config-ip-sla-tcp)# history lives-kept 2
```

(Optional) Sets the number of lives maintained in the history table for an IP SLAs operation.

Step 12 **owner** *owner-id*

Example:

```
Device(config-ip-sla-tcp)# owner admin
```

(Optional) Configures the SNMP owner of an IP SLAs operation.

Step 13 **history lives-kept** *lives*

Example:

```
Device(config-ip-sla-tcp)# history lives-kept 2
```

(Optional) Sets the number of lives maintained in the history table for an IP SLAs operation.

Step 14 **owner** *owner-id*

Example:

```
Device(config-ip-sla-tcp)# owner admin
```

(Optional) Configures the SNMP owner of an IP SLAs operation.

Step 15 **history statistics-distribution-interval** *milliseconds*

Example:

```
Device(config-ip-sla-tcp)# history statistics-distribution-interval 10
```

(Optional) Sets the time interval for each statistics distribution kept for an IP SLAs operation.

Step 16 **tag** *text*

Example:

```
Device(config-ip-sla-tcp)# tag TelnetPollServer1
```

(Optional) Creates a user-specified identifier for an IP SLAs operation.

Step 17 **threshold** *miiliseconds*

Example:

```
Device(config-ip-sla-tcp)# threshold 10000
```

(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.

Step 18 **timeout** *miiliseconds*

Example:

```
Device(config-ip-sla-tcp)# timeout 10000
```

(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.

Step 19 Configure one of the following:

- **tos** *number*
- **traffic-class** *number*

Example:

```
Device(config-ip-sla-tcp)# tos 160
OR
Device(config-ip-sla-tcp)# traffic-class 160
```

Optional) Defines the type of byte in the IPv4 header of an IP SLAs operation.

- **tos** *number*: Defines the ToS byte in the IPv4 header of an IP SLAs operation.
- **traffic-class** *number*: Defines the traffic class byte in the IPv6 header for a supported IP SLAs operation.

Step 20 **flow-label** *number*

Example:

```
Device(config-ip-sla-tcp)# flow-label 112233
```

(Optional) In an IPv6 network only, defines the flow label field in the IPv6 header for a supported IP SLAs operation.

Step 21 **exit**

Example:

```
Device(config-ip-sla-tcp)# exit
```

Exits UDP configuration submode and returns to global configuration mode.

Step 22 **show ip sla configuration** [*operation-number*]

Example:

```
Device# show ip sla configuration 10
```

(Optional) Displays configuration values including all defaults for all IP SLAs operations or a specified operation.

Configuration examples for IP SLAs TCP connect operations

The following example shows how to configure a TCP Connect operation from Device B to the Telnet port (TCP port 23) of IP Host 1 (IP address 10.0.0.1), as shown in the "TCP Connect Operation" figure in the "Information About the IP SLAs TCP Connect Operation" section. The operation is scheduled to start immediately. In this example, the control protocol is disabled on the source (Device B). IP SLAs uses the control protocol to notify the IP SLAs responder to enable the target port temporarily. This action allows the responder to reply to the TCP Connect operation. In this example, because the target is not a Cisco device and a well-known TCP port is used, there is no need to send the control message.

Device A (target device) Configuration

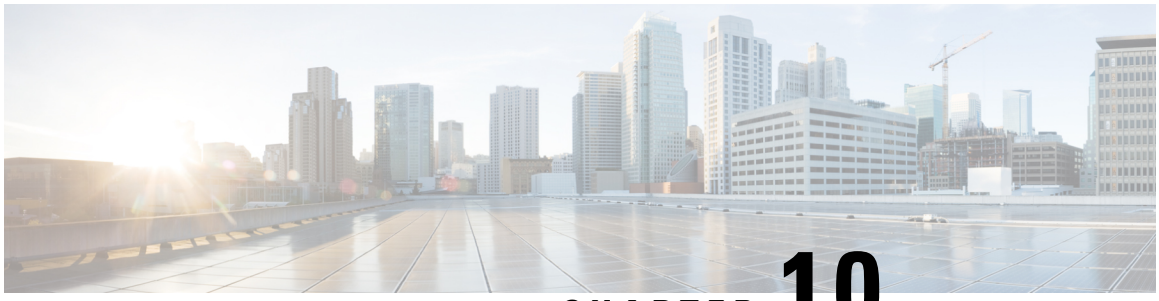
```
Device> enable
Device# configure terminal
Device(config)# ip sla responder tcp-connect ipaddress 10.0.0.1 port 23
```

Device B (source device) Configuration

```
Device> enable
Device# configure terminal
Device(config)# ip sla 9
Device(config-ip-sla)# tcp-connect 10.0.0.1 23 control disable
Device(config-ip-sla-tcp)# frequency 30
Device(config-ip-sla-tcp)# tos 128
Device(config-ip-sla-tcp)# timeout 1000
Device(config-ip-sla-tcp)# tag FLL-RO
Device(config-ip-sla-tcp)# ip sla schedule 9 start-time now
```

The following example shows how to configure a TCP Connect operation with a specific port, port 23, and without an IP SLAs responder. The operation is scheduled to start immediately and run indefinitely.

```
Device> enable
Device# configure terminal
Device(config)# ip sla 9
Device(config-ip-sla)# tcp-connect 173.29.139.132 21 control disable
Device(config-ip-sla-tcp)# frequency 30
Device(config-ip-sla-tcp)# ip sla schedule 9 life forever start-time now
```



CHAPTER 10

IP SLAs UDP Echo Operation

Feature History for IP SLAs - UDP Echo Operation

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	IP SLAs - UDP Echo Operation: This operation measures the response time needed to establish a UDP connection between a Cisco device and a target device across an IP network.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

IP SLAs UDP Echo

The UDP echo operation measures the end-to-end response time between a Cisco device and other IP-enabled devices. Utilizing UDP, a transport layer (Layer 4) Internet protocol commonly used by various IP-based services, this operation sends and receives UDP packets to assess response times and test connectivity across the network. By performing these measurements, network administrators can verify overall network performance and effectively troubleshoot connectivity issues, ensuring reliable service delivery.

The results of a UDP echo operation are valuable for troubleshooting issues with business-critical applications, as they provide information on round-trip delay times and verify connectivity to both Cisco and non-Cisco devices. By analyzing these results, network administrators can identify potential performance bottlenecks and ensure reliable communication throughout the network infrastructure. This capability is essential for maintaining optimal performance and availability of important network services.

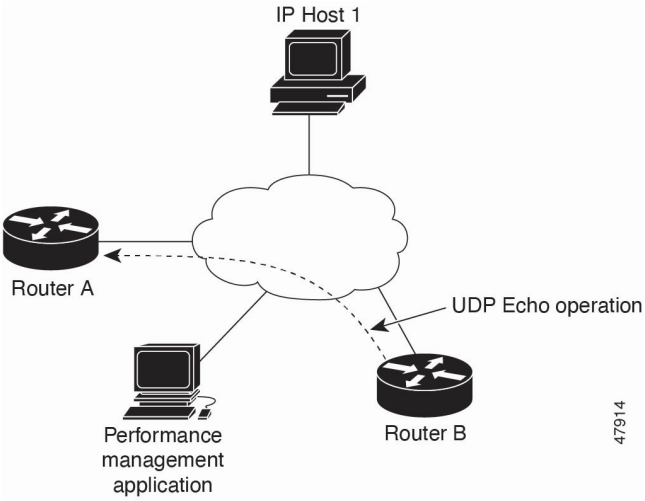
How IP SLA IP SLAs UDP echo works

Summary

In the following figure Device A has been configured as an IP SLAs Responder and Device B is configured as the source IP SLAs device.

Workflow

Figure 19: UDP Echo Operation



Response time, or round-trip time, is calculated by measuring the interval between sending a UDP echo request from Device B to the destination device, Device A, and receiving a UDP echo reply from Device A. The accuracy of the UDP echo operation is improved when the IP SLAs Responder is enabled on Device A, provided it is a Cisco device. For Cisco devices, IP SLAs can send a UDP datagram to any specified port number, and using the IP SLAs Responder is optional for UDP echo operations. However, the IP SLAs Responder feature cannot be configured on non-Cisco devices.

Configure a UDP echo operation on the source device

Follow the steps in each of these tasks to configure a UDP echo operation on the source device

Procedure

Step 1 Perform any one of these tasks:

- Configure a basic UDP echo operation on the source device
- Configure a UDP echo operation with optional parameters on the source device

Step 2 Schedule IP SLAs operations

Configure a basic UDP echo operation on the source device

Perform this task to configure a basic UDP echo operation on the source device.

Before you begin

If you are using the IP SLAs Responder, ensure that you have completed the "Configuring the IP SLAs Responder on the Destination Device" section before you start this task.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla** *operation-number*

Example:

```
Device(config)# ip sla 10
```

Starts configuring an IP SLAs operation and enters IP SLA configuration mode.

Step 4 **udp-echo** {*destination-ip-addres* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}]

Example:

```
Device(config-ip-sla)# udp-echo 172.29.139.134 5000
```

Defines a UDP echo operation and enters IP SLA UDP configuration mode.

Use the **control disable** keyword combination only if you disable the IP SLAs control protocol on both the source and target devices

- *destination-ip-address* | *destination-hostname*: Specifies the IP address or hostname of the target device for the UDP Jitter operation.
- *destination-port*: The UDP port number on the target device that will receive the packets.
- **source-ip** {*ip-address* | *hostname*}: (Optional) Specifies the source IP address or hostname from which packets will be sent.
- **source-port** *port-number*: (Optional) Specifies the UDP source port number for the test packets.
- **control** {**enable** | **disable**}: (Optional) Enables or disables the control protocol, which is used to notify the responder on the target device about the test.

Use the **control disable** keyword combination only if you disable the IP SLAs control protocol on both source and destination devices.

Step 5 **data-pattern** *hex-value*

Example:

```
Device(config-ip-sla-udp)# data-pattern FFFFFFFF
```

(Optional) Sets a hexadecimal value for data pattern.

The range is 0 to FFFFFFFF.

Step 6 **frequency** *seconds*

Example:

```
Device(config-ip-sla)# frequency 30
```

(Optional) Sets the rate at which a specified IP SLAs operation repeats.

Step 7 **end**

Example:

```
Device(config-ip-sla)# end
```

Returns to privileged EXEC mode.

What to do next

To add proactive threshold conditions and reactive triggering for generating traps, or for starting another operation, to an IP SLAs operation, see the "Configuring Proactive Threshold Monitoring" section.

Configure a UDP echo operation with optional parameters on the source device

Perform this task to configure a UDP echo operation with optional parameters on the source device.

Before you begin

If you are using an IP SLAs Responder in this operation, the responder must be configured on the destination device. See the "Configuring the IP SLAs Responder on the Destination Device."

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla** *operation-number*

Example:

```
Device(config)# ip sla 10
```

Starts configuring an IP SLAs operation and enters IP SLA configuration mode.

Step 4 **udp-echo** {*destination-ip-addres* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}]

Example:

```
Device(config-ip-sla)# udp-echo 172.29.139.134 5000
```

Defines a UDP echo operation and enters IP SLA UDP configuration mode.

Use the **control disable** keyword combination only if you disable the IP SLAs control protocol on both the source and target devices

- *destination-ip-address* | *destination-hostname*: Specifies the IP address or hostname of the target device for the UDP Jitter operation.
- *destination-port*: The UDP port number on the target device that will receive the packets.
- **source-ip** {*ip-address* | *hostname*}: (Optional) Specifies the source IP address or hostname from which packets will be sent.
- **source-port** *port-number*: (Optional) Specifies the UDP source port number for the test packets.
- **control** {**enable** | **disable**}: (Optional) Enables or disables the control protocol, which is used to notify the responder on the target device about the test.

  Use the **control disable** keyword combination only if you disable the IP SLAs control protocol on both source and destination devices.

Step 5 **history buckets-kept** *size*

Example:

```
Device(config-ip-sla-udp)# history buckets-kept 25
```

(Optional) Sets the number of history buckets that are kept during the lifetime of an IP SLAs operation.

Step 6 **data-pattern** *hex-pattern*

Example:

```
Device(config-ip-sla-udp)# data-pattern
```

(Optional) Specifies the data pattern in an IP SLAs operation to test for data corruption.

Step 7 **history distributions-of-statistics-kept** *size*

Example:

```
Device(config-ip-sla-udp)# history distributions-of-statistics-kept 5
```

(Optional) Sets the number of statistics distributions kept per hop for an IP SLAs operation.

size: The range is from 1 to 20.

Step 8 **history-enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*]

Example:

```
Device(config-ip-sla-udp)# history enhanced interval 900 buckets 100
```

(Optional) Enables enhanced history gathering for an IP SLAs operation.

- **interval** *seconds*: (Optional) The interval, in seconds, at which to collect and store enhanced statistics. The range is from 1 to 3600 seconds.
- **buckets** *number-of-buckets*: (Optional) The number of enhanced history buckets to retain. The range is from 1 to 100.

Step 9 **history filter** {**none** | **all** | **overThreshold** | **failures**}

Example:

```
Device(config-ip-sla-udp)# history filter failures
```

(Optional) Defines the type of information kept in the history table for an IP SLAs operation.

Step 10 **frequency** *seconds*

Example:

```
Device(config-ip-sla-udp)# frequency 30
```

(Optional) Sets the rate at which a specified IP SLAs operation repeats.

Step 11 **history hours-of-statistics-kept** *hours*

Example:

```
Device(config-ip-sla-udp)# history hours-of-statistics-kept 4
```

(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.

Step 12 **history lives-kept** *lives*

Example:

```
Device(config-ip-sla-udp)# history lives-kept 2
```

(Optional) Sets the number of lives maintained in the history table for an IP SLAs operation.

Step 13 **owner** *owner-id*

Example:

```
Device(config-ip-sla-udp)# owner admin
```

(Optional) Configures the SNMP owner of an IP SLAs operation.

Step 14 **request-data-size** *bytes*

Example:

```
Device(config-ip-sla-udp)# request-data-size 64
```

(Optional) Sets the protocol data size in the payload of an IP SLAs operation's request packet.

Step 15 **history statistics-distribution-interval** *milliseconds*

Example:

```
Device(config-ip-sla-udp)# history statistics-distribution-interval 10
```

(Optional) Sets the time interval for each statistics distribution kept for an IP SLAs operation.

Step 16 **tag** *text*

Example:

```
Device(config-ip-sla-udp)# tag TelnetPollServer1
```

(Optional) Creates a user-specified identifier for an IP SLAs operation.

Step 17 **threshold** *miiliseconds*

Example:

```
Device(config-ip-sla-udp)# threshold 10000
```

(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.

Step 18 **timeout** *miiliseconds*

Example:

```
Device(config-ip-sla-udp)# timeout 10000
```

(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.

Step 19 Configure one of the following:

- **tos** *number*
- **traffic-class** *number*

Example:

```
Device(config-ip-sla-udp)# tos 160
OR
Device(config-ip-sla-udp)# traffic-class 160
```

Optional) Defines the type of byte in the IPv4 header of an IP SLAs operation.

- **tos** *number*: Defines the ToS byte in the IPv4 header of an IP SLAs operation.
- **traffic-class** *number*: Defines the traffic class byte in the IPv6 header for a supported IP SLAs operation.

Step 20 **flow-label** *number*

Example:

```
Device(config-ip-sla-udp)# flow-label 112233
```

(Optional) In an IPv6 network only, defines the flow label field in the IPv6 header for a supported IP SLAs operation.

Step 21 **verify-data**

Example:

```
Device(config-ip-sla-udp)# verify-data
```

(Optional) Causes an IP SLAs operation to check each reply packet for data corruption.

Step 22 **exit**

Example:

```
Device(config-ip-sla-udp)# exit
```

Exits UDP configuration submode and returns to global configuration mode.

What to do next

To add proactive threshold conditions and reactive triggering for generating traps, or for starting another operation, to an IP SLAs operation, see the "Configuring Proactive Threshold Monitoring" section.

Schedule IP SLAs operations

Perform this task to schedule IP SLAs operations.

Before you begin

- All IP SLAs operations to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group must be the same.
- The list of one or more operation ID numbers to be added to a multioperation group must be limited to a maximum of 125 characters in length, including commas (,).

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh:mm:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*}] [**ageout** *seconds*] [**recurring**]

Example:

```
Device(config)# ip sla schedule 10 life forever start-time
OR
Device(config)# ip sla schedule 1 3,4,6-9 schedule-period 50 frequency range 80-100
```

(Optional) Configures the scheduling parameters for an individual IP SLAs operation.

- *operation-number*: The IP SLA operation number to schedule (must match a previously created IP SLA operation).

The range is from 1 to 2147483647.

- **life** {**forever** | *seconds*}: How long the operation will run.
  - **forever**: Runs the operation continuously until manually stopped.
  - *seconds*: Number of seconds the operation should run.

    The range is from 1 to 2147483647 seconds.
- **start-time** {[*hh:mm:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*}: Specifies when to start the operation.
  - *hh:mm:ss* [*month day* | *day month*]: Specific time and date.
  - **pending**: Waits for a manual start.
  - **now**: Starts immediately.
  - **after** *hh:mm:ss*: Starts after the specified amount of time.
- **ageout** *seconds*: Time (in seconds) after which the operation is automatically deleted.

  The range is from 0 to 2147483647 seconds.
- **recurring**: Makes the operation run repeatedly according to its frequency settin

Step 4 **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* |**schedule-together**}[**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm* [*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm*[*:ss*]}]

Example:

```
Device(config)# ip sla group schedule 10 schedule-period frequency
OR
Device(config)# ip sla group schedule 1 3,4,6-9 life forever start-time now
```

(Optional) Specifies an IP SLAs operation group number and the range of operation numbers for a multioperation scheduler.

- *group-operation-number*: The number assigned to the group operation (must be unique).

  The range is from 1 to 2147483647.
- *operation-id-numbers*: List of individual IP SLA operation numbers to be included in the group.

  The range is from 1 to 2147483647 (can be a series separated by spaces).
- **schedule-period** *schedule-period-range*: Schedules each operation in the group with a specified time period between them.

  The range is from 1 to 604800 (seconds; up to 7 days).
- **schedule-together**: Starts all operations in the group at the same time.
- **frequency** *group-operation-frequency*: How often (in seconds) the group operation runs.

  The range is from 1 to 604800 seconds.

Step 5 **end**

Example:

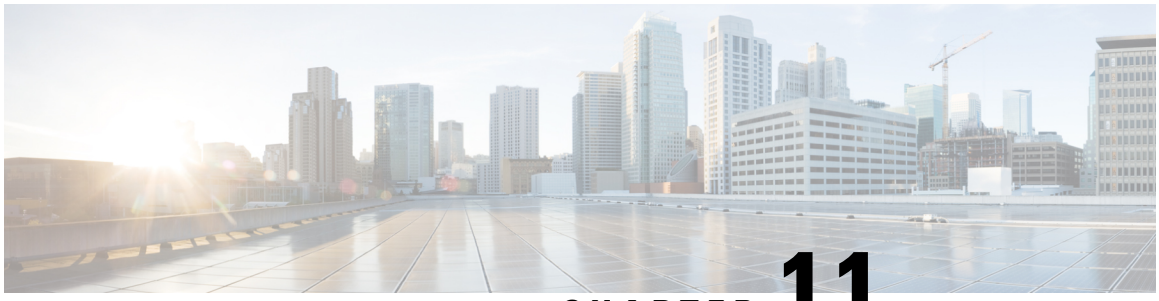
```
Device(config)# end
```

Exits global configuration mode and returns to privileged EXEC mode.

Configuration example for IP SLAs UDP echo operations

The following example configures an IP SLAs operation type of UDP echo that will start immediately and run indefinitely.

```
Device> enable
Device# configure terminal
Device(config)# ip sla 5
Device(config-ip-sla)# udp-echo 172.29.139.134 5000
Device(config-ip-sla-udp)# frequency 30
Device(config-ip-sla-udp)# request-data-size 160
Device(config-ip-sla-udp)# tos 128
Device(config-ip-sla-udp)# timeout 1000
Device(config-ip-sla-udp)# tag FLL-RO
Device(config-ip-sla-udp)# ip sla schedule 5 life forever start-time now
```



CHAPTER 11

IP SLAs UDP Jitter Operation

Feature History for IP SLAs - UDP Jitter Operation

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	IP SLAs - UDP Jitter Operation: This operation assesses whether a network is suitable for real-time traffic applications such as Voice over IP (VoIP), video over IP, or real-time conferencing	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

IP SLAs UDP jitter

The IP Service Level Agreements (SLAs) UDP jitter operation is designed to assess whether a network is suitable for real-time traffic applications such as Voice over IP (VoIP), video over IP, or real-time conferencing. By measuring variations in packet delay and other performance metrics, this operation helps ensure that the network can support the quality requirements necessary for seamless, real-time communication.

Jitter

Jitter refers to the variation in delay between packets as they travel from the source to the destination. For instance, if packets are sent every 10 milliseconds from the source, an ideal network would deliver them to the destination exactly 10 milliseconds apart. However, network factors such as queuing or routing changes

can cause packets to arrive with more or less delay than expected. If a packet arrives 12 milliseconds after the previous one, the positive jitter is 2 milliseconds; if it arrives 8 milliseconds after, the negative jitter is 2 milliseconds. For applications that are sensitive to delays, like VoIP, positive jitter is problematic because it disrupts the steady flow of data, while a jitter value of zero is considered optimal for smooth, real-time communication.

How IP SLAs UDP jitter works

Summary

The UDP jitter operation works by generating synthetic, or simulated, UDP traffic between network devices. This operation supports asymmetric probes, allowing different packet sizes to be sent in each direction—for example, request packets from the source device to the destination device can be a different size than response packets traveling back. During the test, the source device sends a specified number (N) of UDP packets, each with a defined size (S), spaced T milliseconds apart, and this test is repeated at a set frequency (F). The destination device responds with UDP packets of another specified size (P). By default, the operation sends ten UDP packets (N), each with a 10-byte payload (S), every 10 milliseconds (T), and repeats the entire process every 60 seconds (F). All these parameters are customizable, allowing you to closely simulate the real IP services and traffic patterns you want to monitor in your network.

Table 2: UDP Jitter Operation Parameters

UDP jitter operation parameter	Default	Configuration commands
Number of packets (N)	10 packets	**udp-jitter num-packets**
Payload size per request packet (S)	10 bytes	**request-data-size**
Payload size per response packet (P)	The default response data size varies depending on the type of IP SLAs operation configured. **Note** If the **response-data-size** command is not configured, then the response data size value is the same as the request data size value.	**response-data-size**
Time between packets, in milliseconds (T)	10 ms	**udp-jitter interval**
Elapsed time before the operation repeats, in seconds (F)	60 seconds	**frequency** (IP SLA)

IP SLAs operations work by generating synthetic, or simulated, network traffic to test and monitor network performance. Each IP SLA operation, such as operation 10, is configured to run repeatedly at a specified frequency for as long as the operation is active. This ongoing process allows continuous monitoring and helps ensure that network performance is consistently measured over time.

Benefits of IP SLAs UDP jitter

The IP SLAs UDP jitter operation is not limited to simply monitoring jitter; it serves as a versatile data-gathering tool for network performance. The UDP jitter operation collects detailed information by including packet sequence numbers and sending and receiving time stamps for both the source and the operational target. This allows it to measure several important metrics, such as per-direction jitter (in both directions), per-direction packet loss, per-direction delay (one-way delay), and round-trip delay (average round-trip time). Since network paths can be asymmetric—with data taking different routes in each direction—having per-direction statistics helps network administrators pinpoint exactly where congestion or other issues are occurring within the network.

Guidelines to configure IP SLAs UDP jitter

- Time synchronization, such as that provided by the Network Time Protocol (NTP), is required between the source and target device to provide accurate one-way delay (latency) measurements.
- Time synchronization is not required for one-way jitter and packet loss measurements. If time is not synchronized between source and target devices, one-way jitter and packet loss data are still returned, but the one-way delay measurements provided by the UDP jitter operation will have a value of "0".
- Before configuring any IP Service Level Agreements (SLAs) application, use the **show ip sla application** command to verify that the operation type is supported on the software image.
- Multiple SLA probes configured with the same source and destination IP addresses and port numbers should not be run simultaneously.
- The IP SLAs UDP jitter operation does not support the IP SLAs History feature due to the large volume of data involved. As a result, the commands **history buckets-kept**, **history filter**, **history lives-kept**, **samples-of-history-kept**, and **show ip sla history** are not supported for UDP jitter operations.
- The MIB used by IP SLAs (CISCO-RTTMON-MIB) restricts the hours-of-statistics kept for the UDP jitter operation to a maximum of two hours. Configuring a higher value with the **history hours-of-statistics** *hours* command does not extend this limit. However, historical data for the operation can be collected using the Data Collection MIB. For more details, refer to the CISCO-DATA-COLLECTION-MIB.
- If the IP SLAs operation is not running and not generating statistics, adding the **verify-data** command in IP SLA configuration mode enables data verification. With data verification enabled, each operation response is checked for corruption. However, the **verify-data command** should be used with caution during normal operations, as it can generate unnecessary overhead.
- Use the **debug ip sla trace** and **debug ip sla error** commands to help troubleshoot issues with an IP SLAs operation.

Configure and schedule a UDP jitter operation on a source device

Follow the steps in each of these tasks to configure and schedule a UDP jitter operation on a source device.

Procedure

Step 1 Perform any one of these tasks:

- Configure a basic UDP jitter operation on a source device
- ConfigConfigure a UDP jitter operation with additional characteristics, on page 125

Step 2 Schedule IP SLAs operations

Configure a basic UDP jitter operation on a source device

Perform this task to configure a basic UDP jitter operation on a source device.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla** *operation-number*

Example:

```
Device(config)# ip sla 10
```

Starts configuring an IP SLAs operation and enters IP SLA configuration mode.

Step 4 **udp-jitter** {*destination-ip-addres* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}] [**num-packets** *number-of-packets*] [**interval** *interpacket-interval*]

Example:

```
Device(config-ip-sla)# udp-jitter 192.0.2.135 5000
```

Configures the IP SLAs operation as a UDP jitter operation and enters UDP jitter configuration mode.

- *destination-ip-address* | *destination-hostname*: Specifies the IP address or hostname of the target device for the UDP Jitter operation.
- *destination-port*: The UDP port number on the target device that will receive the packets.

- **source-ip** {*ip-address* | *hostname*}: (Optional) Specifies the source IP address or hostname from which packets will be sent.
- **source-port** *port-number*: (Optional) Specifies the UDP source port number for the test packets.
- **control** {**enable** | **disable**}: (Optional) Enables or disables the control protocol, which is used to notify the responder on the target device about the test.

  Use the **control disable** keyword combination only if you disable the IP SLAs control protocol on both source and destination devices.
- **num-packets** *number-of-packets*: (Optional) Sets the number of UDP packets to send in each test.
- **interval** *interpacket-interval*: (Optional) Sets the interval (in milliseconds) between sending each packet.

Step 5 **frequency** *seconds*

Example:

```
Device(config-ip-sla)# frequency 30
```

(Optional) Sets the rate at which a specified IP SLAs operation repeats.

Step 6 **end**

Example:

```
Device(config-ip-sla)# end
```

Exits UDP Jitter configuration mode and returns to privileged EXEC mode.

Step 7 **show ip sla configuration** [*operation-number*]

Example:

```
Device# show ip sla configuration 10
```

(Optional) Displays configuration values including all

defaults for all IP SLAs operations or a specified operation.

Configure a UDP jitter operation with additional characteristics

Before you begin

Before configuring a UDP jitter operation on a source device, the IP SLAs Responder must be enabled on the target device (the operational target). The IP SLAs Responder is available only on Cisco software-based devices.

Perform this task to configure a UDP jitter operation with additional characteristics.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla** *operation-number*

Example:

```
Device(config)# ip sla 10
```

Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.

Step 4 **udp-jitter** {*destination-ip-addres* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}] [**num-packets** *number-of-packets*] [**interval** *interpacket-interval*]

Example:

```
Device(config-ip-sla)# udp jitter 192.0.2.134 5000
```

Configures the IP SLAs operation as a UDP jitter operation and enters UDP jitter configuration mode.

- *destination-ip-address* | *destination-hostname*: Specifies the IP address or hostname of the target device for the UDP Jitter operation.
- *destination-port*: The UDP port number on the target device that will receive the packets.
- **source-ip** {*ip-address* | *hostname*}: (Optional) Specifies the source IP address or hostname from which packets will be sent.
- **source-port** *port-number*: (Optional) Specifies the UDP source port number for the test packets.
- **control** {**enable** | **disable**}: (Optional) Enables or disables the control protocol, which is used to notify the responder on the target device about the test.

  Use the **control disable** keyword combination only if you disable the IP SLAs control protocol on both source and destination devices.
- **num-packets** *number-of-packets*: (Optional) Sets the number of UDP packets to send in each test.
- **interval** *interpacket-interval*: (Optional) Sets the interval (in milliseconds) between sending each packet.

Step 5 **history distributions-of-statistics-kept** *size*

Example:

```
Device(config-ip-sla-jitter)# history distributions-of-statistics-kept 5
```

(Optional) Sets the number of statistics distributions kept per hop for an IP SLAs operation.

size: The range is from 1 to 20.

Step 6 **history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*]

Example:

```
Device(config-ip-sla-jitter)# history enhanced interval 900 buckets 100
```

(Optional) Enables enhanced history gathering for an IP SLAs operation.

- **interval** *seconds*: (Optional) The interval, in seconds, at which to collect and store enhanced statistics. The range is from 1 to 3600 seconds.
- **buckets** *number-of-buckets*: (Optional) The number of enhanced history buckets to retain. The range is from 1 to 100.

Step 7 **frequency** *seconds*

Example:

```
Device(config-ip-sla-jitter)# frequency 30
```

(Optional) Sets the rate at which a specified IP SLAs operation repeats.

Step 8 **history hours-of-statistics-kept** *hours*

Example:

```
Device(config-ip-sla-jitter)# history hours-of-statistics-kept 4
```

(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.

The range is from 0 to 25.

Step 9 **owner** *owner-id*

Example:

```
Device(config-ip-sla-jitter)# owner admin
```

(Optional) Configures the SNMP owner of an IP SLAs operation.

Step 10 **request-data-size** *bytes*

Example:

```
Device(config-ip-sla-jitter)# request-data-size 64
```

(Optional) Sets the protocol data size in the payload of an IP SLAs operation request packet.

bytes: The range is from 0 to 5000.

Step 11 **response-data-size** *bytes*

Example:

```
Device(config-ip-sla-jitter)# response-data-size 25
```

(Optional) Sets the protocol data size in the payload of an IP SLAs operation response packet.

bytes: The range is from 0 to 5000.

Step 12 **history statistics-distribution-interval***milliseconds*

Example:

```
Device(config-ip-sla-jitter)# history statistics-distribution-interval 10
```

(Optional) Sets the time interval for each statistics distribution kept for an IP SLAs operation.

Step 13 **tag** *text*

Example:

```
Device(config-ip-sla-jitter)# tag TelnetPollServer1
```

(Optional) Creates a user-specified identifier for an IP SLAs operation.

Step 14 **threshold** *milliseconds*

Example:

```
Device(config-ip-sla-jitter)# threshold 1000
```

(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.

Step 15 **timeout** *milliseconds*

Example:

```
Device(config-ip-sla-jitter)# timeout 1000
```

(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.

Step 16 Choose one of the following:

- **tos** *number*
- **traffic-class** *number*

Example:

```
Device(config-ip-sla-jitter)# tos 160
OR
Device(config-ip-sla-jitter)# traffic-class 160
```

(Optional) Defines the type of byte in the IPv4 header of an IP SLAs operation.

- **tos** *number*: Defines the ToS byte in the IPv4 header of an IP SLAs operation.
- **traffic-class** *number*: Defines the traffic class byte in the IPv6 header for a supported IP SLAs operation.

Step 17 **flow-label** *number*

Example:

```
Device(config-ip-sla-jitter)# flow-label 112233
```

(Optional) Defines the flow label field in the IPv6 header for a supported IP SLAs operation.

Step 18 **verify-data**

Example:

```
Device(config-ip-sla-jitter)# verify-data
```

(Optional) Causes an IP SLAs operation to check each reply packet for data corruption.

Step 19 **vrf** *vrf-name*

Example:

```
Device(config-ip-sla-jitter)# vrf vpn-A
```

(Optional) Allows monitoring within MPLS VPNs using IP SLAs operations.

Step 20 **end**

Example:

```
Device(config-ip-sla-jitter)# end
```

Exits UDP jitter configuration mode and returns to privileged EXEC mode.

Step 21 **show ip sla configuration** [*operation-number*]

Example:

```
Device# how ip sla configuration 10
```

(Optional) Displays configuration values including all defaults for all IP SLAs operations or a specified operation.

What to do next

To configure the percentile option for your operation, see the "Configuring the IP SLAs—Percentile Support for Filtering Outliers" module.

Schedule IP SLAs operations

Perform this task to schedule IP SLAs operations.

Before you begin

- All IP SLAs operations to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group must be the same.
- The list of one or more operation ID numbers to be added to a multioperation group must be limited to a maximum of 125 characters in length, including commas (,).

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh:mm:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*}] [**ageout** *seconds*] [**recurring**]

Example:

```
Device(config)# ip sla schedule 10 life forever start-time
OR
Device(config)# ip sla schedule 1 3,4,6-9 schedule-period 50 frequency range 80-100
```

(Optional) Configures the scheduling parameters for an individual IP SLAs operation.

- *operation-number*: The IP SLA operation number to schedule (must match a previously created IP SLA operation).

  The range is from 1 to 2147483647.

- **life** {**forever** | *seconds*}: How long the operation will run.
  - **forever**: Runs the operation continuously until manually stopped.
  - *seconds*: Number of seconds the operation should run.

    The range is from 1 to 2147483647 seconds.

- **start-time** {[*hh:mm:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*}: Specifies when to start the operation.
  - *hh:mm:ss* [*month day* | *day month*]: Specific time and date.
  - **pending**: Waits for a manual start.
  - **now**: Starts immediately.
  - **after** *hh:mm:ss*: Starts after the specified amount of time.

- **ageout** *seconds*: Time (in seconds) after which the operation is automatically deleted.

  The range is from 0 to 2147483647 seconds.

- **recurring**: Makes the operation run repeatedly according to its frequency settin

Step 4 **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* |**schedule-together**}[**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm* [*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm*[*:ss*]}]

Example:

```
Device(config)# ip sla group schedule 10 schedule-period frequency
OR
Device(config)# ip sla group schedule 1 3,4,6-9 life forever start-time now
```

(Optional) Specifies an IP SLAs operation group number and the range of operation numbers for a multioperation scheduler.

- *group-operation-number*: The number assigned to the group operation (must be unique).

  The range is from 1 to 2147483647.

- *operation-id-numbers*: List of individual IP SLA operation numbers to be included in the group.

  The range is from 1 to 2147483647 (can be a series separated by spaces).

- **schedule-period** *schedule-period-range*: Schedules each operation in the group with a specified time period between them.

  The range is from 1 to 604800 (seconds; up to 7 days).

- **schedule-together**: Starts all operations in the group at the same time.
- **frequency** *group-operation-frequency*: How often (in seconds) the group operation runs.

  The range is from 1 to 604800 seconds.

Step 5 **end**

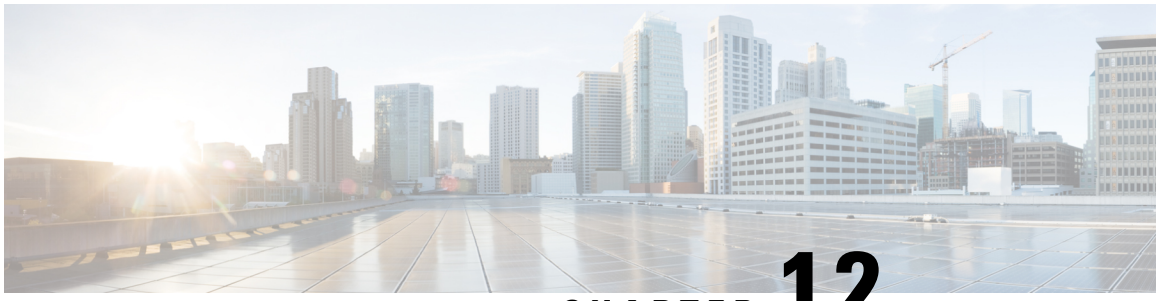
Example:

```
Device(config)# end
```

Exits global configuration mode and returns to privileged EXEC mode.

Verify IP SLAs UDP jitter operations

Command	Purpose
show ip sla configuration	Displays the current configuration details of all IP SLA operations on a Cisco device. This command provides information such as the operation type, target addresses, frequency, timeout values, and other parameters set for each IP SLA operation.
show ip sla group schedule	Displays the scheduling details and status of all IP SLA group operations configured on a Cisco device This includes information such as the group operation number, the member operation IDs included in each group, the scheduling method (such as schedule-together or with a period), start time, frequency, operation life, and current state (active, pending, etc.).
show ip sla statistics	Displays real-time statistical data for all configured IP SLA operations on a Cisco device. This output includes key performance metrics such as operation type, round-trip time (RTT), packet loss, success or failure counts, and the time of the last operation.
show ip sla statistics 2 details	Displays detailed, real-time statistical information for IP SLA operation number 2 on a Cisco device This detailed output includes metrics such as operation type, destination address, round-trip time (RTT), packet loss, jitter, number of successful and failed operations, return codes, and the exact time of the last operation



CHAPTER 12

IP SLAs Reaction Threshold

Feature History for IP SLAs - Reaction Threshold

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	IP SLAs - Reaction Threshold: This feature monitors a specified value or event and intiates a trigger when the defined threshold is exceeded or when a specific event, such as a timeout or connection loss occurs.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

IP SLAs reaction

IP SLA reactions are configured to trigger when a monitored value exceeds or falls below a specified threshold, or when a specific event occurs, such as a timeout or connection loss. If an IP SLA operation detects that a monitored value is either too high or too low according to its configured reactions, it can generate a notification to a network management application or initiate another IP SLA operation to collect additional data.

When an IP SLA operation is triggered, the target operation begins and runs independently, without awareness of the status of the original triggering operation. The target operation will continue to run until a

condition-cleared event occurs. After this event, the target operation stops gracefully, and its state changes from Active to Pending, allowing it to be triggered again.

Supported reactions by IP SLAs operation

The tables below list which reactions are supported for each IP SLA operation.

Table 3: Supported Reaction Configuration, by IP SLA Operation

Reaction	ICMP Echo	Path Echo	UDP Jitter	UDP Echo	TCP Connect	DHCP	DLSW	ICMP Jitter	DNS	Frame Relay
Failure	Y	--	Y	Y	Y	Y	--	Y	Y	--
RTT	Y	Y	--	Y	Y	Y	Y	--	Y	Y
RTTAvg	--	--	Y	--	--	--	--	Y	--	--
timeout	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
connectionLoss	--	--	Y	Y	Y	--	--	--	--	
verifyError	--	--	Y	Y	--	--	--	Y	--	Y
jitterSDAvg	--	--	Y	--	--	--		Y	--	--
jitterAvg	--	--	Y	--	--	--	--	Y	--	--
packetLateArrival	--	--	Y	--	--	--	--	Y	--	--
packetOutOfSequence	--	--	Y	--	--	--	--	Y	--	--
MaxOfPostiveSD	--	--	Y	--	--	--		Y	--	--
MaxOfNegativeSD	--	--	Y	--	--	--	--	Y	--	--
MaxOfPostiveDS	--	--	Y	--	--	--	--	Y	--	--
MaxOfNegativeDS	--	--	Y	--	--	--	--	Y	--	--
MOS	--	--	Y	--	--	--		--	--	--
ICPIF	--	--	Y	--	--	--	--	--	--	--
PacketLossDS	--	--	Y	--	--	--	--	--	--	--
PacketLossSD	--	--	Y	--	--	--	--	--	--	--
PacketMIA	--	--	Y	--	--	--		--	--	--
iaJitterDS	--	--	--	--	--	--	--	--	--	--
frameLossDS	--	--	--	--	--	--	--	--	--	--
mosLQDSS	--	--	--	--	--	--	--	--	--	--

Reaction	ICMP Echo	Path Echo	UDP Jitter	UDP Echo	TCP Connect	DHCP	DLSW	ICMP Jitter	DNS	Frame Relay
mosCQDS	--	--	--	--	--	--	--	--	--	--
rfactorDS	--	--	--	--	--	--	--	--	--	--
iaJitterSD	--	--	--	--	--	--	--	--	--	--
successivePacketLoss	--	--	--	--	--	--	--	Y	--	--
MaxOfLatencyDS	--	--	--	--	--	--	--	Y	--	--
MaxOfLatencySD	--	--	--	--	--	--	--	Y	--	--
LatencyDS	--	--	--	--	--	--	--	Y	--	--
LatencySD	--	--	--	--	--	--	--	Y	--	--
packetLoss	--	--	--	--	--	--	--	Y	--	--

Table 4: Supported Reaction Configuration, by IP SLA Operation

Reaction	HTTP	SLM	RTP	FTP	Lsp Trace	Post delay	Path Jitter	LSP Ping	Gatekeeper Registration
Failure	--	--	--	--	--	--	--	--	--
RTT	Y	Y	Y	Y	Y	Y	Y	Y	Y
RTTAvg	--	--	--	--	--	--	--	--	--
timeout	Y	Y	Y	Y	--	Y	Y	Y	Y
connectionLoss	Y		Y	Y	Y	--	--	Y	--
verifyError	--	--	--	--	--	--	--	--	--
jitterSDAvg	--	--	--	--	--	--	Y	--	--
jitterAvg	--	--	--	--	--	--	Y	--	--
packetLateArrival	--	--	--	--	--	--	Y	--	--
packetOutOfSequence	--	--	--	--	--	--	Y	--	--
MaxOfPostiveSD	--	--	--	--	--	--	Y	--	--
MaxOfNegativeSD	--	--	--	--	--	--	Y	--	--
MaxOfPostiveDS	--	--	--	--	--	--	Y	--	--
MaxOfNegativeDS	--	--	--	--	--	--	Y	--	--
MOS	--	--	--	--	--	--	--	--	--

Reaction	HTTP	SLM	RTP	FTP	Lsp Trace	Post delay	Path Jitter	LSP Ping	Gatekeeper Registration
ICPIF	--	--	--	--	--	--	--	--	--
PacketLossDS	--	--	Y	--	--	--	--	--	--
PacketLossSD	--	--	Y	--	--	--	--	--	--
PacketMIA	--	--	Y	--	--	--	--	--	--
iaJitterDS	--	--	Y	--	--	--	--	--	--
frameLossDS	--	--	Y	--	--	--	--	--	--
mosLQDSS	--	--	Y	--	--	--	--	--	--
mosCQDS	--	--	Y	--	--	--	--	--	--
rfactorDS	--	--	Y						
iaJitterSD	--	--	Y	--	--	--	--	--	--
successivePacketLoss	--	--	--	--	--	--	--	--	--
MaxOfLatencyDS	--	--	--	--	--	--	--	--	--
MaxOfLatencySD	--	--	--	--	--	--	--	--	--
LatencyDS	--	--	--	--	--	--	--	--	--
LatencySD	--	--	--	--	--	--	--	--	--
packetLoss	--	--	--	--	--	--	--	--	--

IP SLAs reaction threshold monitoring and notifications

IP SLAs support proactive threshold monitoring and notifications for performance parameters such as average jitter, unidirectional latency, bidirectional round-trip time (RTT), and connectivity across most IP SLA operations. This proactive monitoring capability also enables the configuration of reaction thresholds for key VoIP-related parameters, including unidirectional jitter, unidirectional packet loss, and unidirectional VoIP voice quality scoring.

Notifications for IP SLAs are configured as triggered reactions. Packet loss, jitter, and Mean Opinion Score (MOS) statistics are specific to IP SLA jitter operations. Notifications can be generated for threshold violations in either direction—source-to-destination or destination-to-source—or for out-of-range RTT values related to packet loss and jitter. Events such as traps are triggered when the RTT value exceeds or falls below a specified threshold.

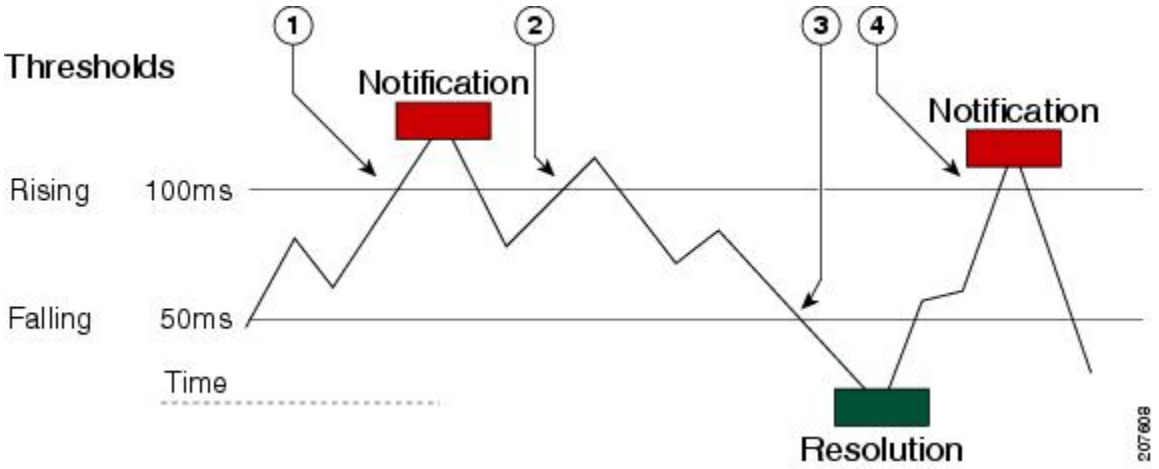
IP SLAs can generate system logging (syslog) messages when a reaction condition occurs. These syslog messages can also be sent as Simple Network Management Protocol (SNMP) traps (notifications) using the CISCO-RTTMON-MIB. SNMP traps for IP SLAs are supported by both the CISCO-RTTMON-MIB and the CISCO-SYSLOG-MIB.

Severity levels in the CISCO-SYSLOG-MIB are defined as follows: SyslogSeverity INTEGER {emergency(1), alert(2), critical(3), error(4), warning(5), notice(6), info(7), debug(8)}. However, severity levels for the system logging process in Cisco software are defined differently: {emergency (0), alert (1), critical (2), error (3), warning (4), notice (5), informational (6), debugging (7)}.

IP SLA threshold violations are logged as level 6 (informational) within the Cisco system logging process, but are sent as level 7 (info) traps from the CISCO-SYSLOG-MIB.

Notifications are not issued for every occurrence of a threshold violation. The process works as follows: when the monitored element exceeds the upper (rising) threshold for the first time, an event is sent and a notification is issued. Subsequent notifications are only generated after the monitored value falls below the lower (falling) threshold and then exceeds the upper threshold again.

Figure 20: IP SLAs triggered teaction condition and notifications for threshold exceeded



1	An event is sent and a threshold-exceeded notification is issued when the rising threshold is exceeded for the first time.
2	Consecutive over-rising threshold violations occur without issuing additional notifications.
3	Themonitored value goes below the falling threshold.
4	Another threshold-exceeded notification is issued when the rising threshold is exceeded only after the monitored value first fell below the falling threshold.

Note A lower-threshold notification is also issued the first time the monitored element falls below the falling threshold. As described, subsequent notifications for lower-threshold violations are only generated after the monitored value first exceeds the rising threshold and then falls below the falling threshold again.

RTT Reactions for jitter operations

RTT reactions for jitter operations are triggered only at the end of the operation and use the most recent value for return-trip time (LatestRTT), which is equal to the average return-trip time (RTTAvg).

SNMP traps for RTT in jitter operations are based on the average return-trip time (RTTAvg) for the entire operation and do not include RTT values for individual packets sent during the operation. For example, if the average RTT is below the threshold, it is possible for up to half of the packets to have RTT values above the threshold. However, this detail is not reflected in the notification, as only the overall average is reported.

Only syslog messages are supported for RTTAvg threshold violations. These syslog messages are sent from the CISCO-RTTMON-MIB.

Guidelines to configure reaction threshold

- RTT reactions for jitter operations are triggered only at the end of the operation and use the latest value for the return-trip time (LatestRTT).
- SNMP traps for RTT for jitter operations are based on the average value for the return-trip time (RTTAvg) for the whole operation only and do not include return-trip time values for individual packets sent during the operation. Only syslog messages are supported for RTTAvg threshold violations.
- Only syslog messages are supported for RTT violations during Jitter operations.
- Only SNMP traps are supported for RTT violations during non-Jitter operations.
- Only syslog messages are supported for non-RTT violations other than timeout, connectionLoss, or verifyError.
- Both SNMP traps and syslog messages are supported for timeout, connectionLoss, or verifyError violations only

Configure IP SLAs Reaction Threshold

Before you begin

IP SLAs operations to be started when violation conditions are met must be configured.

Perform this task to configure thresholds and reactive triggering for generating traps or starting another operation.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla reaction-configuration** *operation-number* **react** *monitored-element* [**action-type** *option*] [**threshold-type** {**average** [*number-of-measurements*] | **consecutive** [*occurrences*] | **immediate** | **never** | **xofy** [*x-value y-value*]}] [**threshold-value** *upper-threshold lower-threshold*]

Example:

```
Device(config)# ip sla reaction-configuration 10 react jitterAvg threshold-type immediate
threshold-value 5000 3000 action-type trapAndTrigger
```

Configures the action (SNMP trap or IP SLAs trigger) that is to occur based on violations of specified thresholds.

- *operation-number*: The number of the IP SLA operation to which the reaction will be applied.
- *monitored-element*: The specific parameter to monitor (e.g., rtt, jitter, packet-loss).
- **action-type** *option*: The action to take when the threshold is crossed (optional, e.g., trapOnly, triggerOnly, trapAndTrigger).
- **threshold-type**: The method for evaluating the threshold. Options include:
  - **average** *number-of-measurements*: Triggers based on the average over a specified number of measurements.
  - **consecutive** *occurrences*: Triggers after a specified number of consecutive occurrences.
  - **immediate**: Triggers immediately when the threshold is crossed.
  - **never**: Disables reactions.
  - **xofy** [*x-value y-value*]: Triggers when x out of y occurrences exceed the threshold.
  - **threshold-value** *upper-threshold lower-threshold*: Specifies the upper and lower threshold values for the monitored element.

Step 4 **ip sla reaction-trigger** *operation-number target-operations*

Example:

```
Device(config)# ip sla reaction-trigger 10 2
```

(Optional) Starts another IP SLAs operation when the violation conditions are met.

This command is required only if the **ip sla reaction-configuration** command is configured with either the **trapAndTrigger** or **triggerOnly** keyword.

Step 5 **ip sla logging traps**

Example:

```
Device(config)# ip sla logging traps
```

(Optional) Enables IP SLAs syslog messages from CISCO-RTTMON-MIB.

Step 6 Configure one of the following:

- **snmp-server enable traps rtr**
- **snmp-server enable traps syslog**

Example:

```
Device(config)# snmp-server enable traps rtr
OR
Device(config)# snmp-server enable traps syslog
```

Enables the system to generate SNMP traps.

The first example shows how to enable the system to generate CISCO-RTTMON-MIB. The second example shows how to enable the system to generate CISCO-SYSLOG-MIB traps.

Step 7 **snmp-server host** {*hostname* | *ip-address*} [**vrf** *vrf-name*] [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*]

Example:

```
Device(config)# snmp-server host 10.1.1.1 public syslog
```

(Optional) Sends traps to a remote host.

Required if the **snmp-server enable traps** command is configured.

Step 8 **exit**

Example:

```
Device(config)# exit
```

Exits global configuration mode and returns to privileged EXEC mode.

Step 9 **show ip sla reaction-configuration** [*operation-number*]

Example:

```
Device# show ip sla reaction-configuration 10
```

(Optional) Displays the configuration of proactive threshold monitoring.

Step 10 **show ip sla reaction-trigger** [*operation-number*]

Example:

```
Device# show ip sla reaction-trigger 2
```

(Optional) Displays the configuration status and operational state of target operations to be triggered.

Configuration examples for IP SLAs reaction threshold

The following example shows how to configure IP SLAS reaction threshold using the **ip sla reaction-configuration** command. In this example, IP SLAs operation 10 is configured to send an SNMP logging trap when the MOS value either exceeds 4.9 (best quality) or falls below 2.5 (poor quality):

```
Device> enable
Device# configure terminal
Device(config)# ip sla reaction-configuration 10 react mos threshold-type immediate
threshold-value 490 250 action-type trapOnly
```

The following example is a sample output of the **show ip sla reaction-configuration** command.

```
Device# show ip sla reaction-configuration 1
Entry number: 1
Reaction Configuration not configured
```

The following example show the default configuration of the **ip sla reaction-configuration** command.

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# ip sla reaction-configuration 1
Device(config)# do show ip sla reaction-configuration 1
Entry number: 1
Reaction: rtt
Threshold Type: Never
Rising (milliseconds): 5000
Falling (milliseconds): 3000
Threshold Count: 5
Threshold Count2: 5
Action Type: None
```

The following example shows that multiple monitored elements are configured for the IP SLAs operation (1), as indicated by the values of Reaction: in the output:

```
Device# show ip sla reaction-configuration
Entry Number: 1
Reaction: RTT
Threshold type: Never
Rising (milliseconds): 5000
Falling (milliseconds): 3000
Threshold Count: 5
Threshold Count2: 5
Action Type: None
Reaction: jitterDSAvg
Threshold type: average
Rising (milliseconds): 5
Falling (milliseconds): 3
Threshold Count: 5
Threshold Count2: 5
Action Type: triggerOnly
Reaction: jitterDSAvg
Threshold type: immediate
Rising (milliseconds): 5
Falling (milliseconds): 3
Threshold Count: 5
Threshold Count2: 5
Action Type: trapOnly
Reaction: PacketLossSD
Threshold type: immediate
Rising (milliseconds): 5
Threshold Falling (milliseconds): 3
Threshold Count: 5
Threshold Count2: 5
Action Type: trapOnly
```

The following example shows how to configure IP SLAs reaction threshold so that CISCO-SYSLOG-MIB traps are sent to the remote host at 10.1.1.1 if the threshold values for RTT or VoIP MOS are violated:

```
! Configure the operation on source.
Device(config)# ip sla 1
Device(config-ip-sla)# udp-jitter 10.1.1.1 3000 codec g711alaw
Device(config-ip-sla-jitter)# exit
Device(config)# ip sla schedule 1 start now life forever
! Configure thresholds and reactions.
Device(config)# ip sla reaction-configuration 1 react rtt threshold-type immediate
threshold-value 3000 2000 action-type trapOnly
Device(config)# ip sla reaction-configuration 1 react MOS threshold-type consecutive 4
threshold-value 390 220 action-type trapOnly
Device(config)# ip sla logging traps
```

```
! The following command sends traps to the specified remote host.
Device(config)# snmp-server host 10.1.1.1 version 2c public syslog
! The following command is needed for the system to generate CISCO-SYSLOG-MIB traps.
Device(config)# snmp-server enable traps syslog
```

The following sample system logging messages shows that IP SLAs threshold violation notifications are generated as level 6 (informational) in the Cisco system logging process:

```
3d18h:%RTT-6-SAATHRESHOLD:RTR(11):Threshold exceeded for MOS
```

This following sample SNMP notification from the CISCO-SYSLOG-MIB for the same violation is a level 7 (info) notification:

```
3d18h:SNMP:V2 Trap, reqid 2, errstat 0, erridx 0
sysUpTime.0 = 32613038
snmpTrapOID.0 = ciscoSyslogMIB.2.0.1
clogHistoryEntry.2.71 = RTT
clogHistoryEntry.3.71 = 7
clogHistoryEntry.4.71 = SAATHRESHOLD
clogHistoryEntry.5.71 = RTR(11):Threshold exceeded for MOS
clogHistoryEntry.6.71 = 32613037
```